WDM Network Design and Destination Conflicts

Victor Yau

September 20, 1996

A thesis submitted for the degree of
PhD in Computer Science
in the
University of Canterbury

Abstract

The parallel use of multiple channels in a WDM star network means that too many packets may simultaneously arrive for the same destination station, necessitating the implementation of a destination-conflict-resolution function somewhere within the network. This thesis considers explicitly, the *placement* of the destination-conflict-resolution function which specifies the location(s) *where* it should be performed, and *when* it should be performed. Traditional placements in which the function is located at all user stations and performed either *before* packet transmission (using the request-schedule-then-transmit principle) or *after* a destination conflict has been detected (using the detect-and-retransmit-if-lost principle), is compared with a central placement in which only one central station located at the entrance to the star coupler is responsible for detecting conflicts and re-scheduling the arrival times of "otherwise lost" packets whilst they are *en route* to their destinations, so that they arrive when their destinations free to receive them.

The networks are evaluated considering their delay and throughput characteristics, the computational complexity of their protocols, and their hardware demands. All numerical results were produced using AKAROA an object-oriented parallel simulation package developed by us for automated precision control of steady-state estimates and automated parallel execution of quantitative simulations. The results presented suggests that significant performance improvements are achievable with the central placement since destination conflicts are resolved without having to retransmit packets nor waiting until the end of a request-broadcast-and-schedule phase before a given packet can be transmitted. The central station works with Space Division Multiplexed signals, just before they enter the star coupler. Its implementation is therefore simpler than when all stations are charged with this task, each of which has to attend to multiple WDM channels. Only "otherwise lost" packets are buffered so the network has low buffer memory requirements.

Acknowledgements

I wish to acknowledge the role of Associate Professor Krzysztof Pawlikowski, who is the supervisor of this thesis. I am fortunate have a supervisor with the kindest of intentions, giving me welcomed encouragement and bringing me many valuable opportunties throughout the course of my PhD studies.

I am deeply grateful to Dr. Marshall Bowen (Department of Computer Science, Western Illinois University, Macomb USA) for introducing me to the area of performance analysis and simulation, and for writing some of the simulator programs, and providing many ideas and insightful comments on my work in the related area of multiple-bus processor-memory interconnection networks, during my early postgraduate years. I am also indebted to Mr. Kevin Watts (Information Technology Services Center, Upper Hutt Wellington, N.Z.), for his kind support and for providing the study leave which made my postgraduate studies possible.

In addition, thanks are due to Professor Anthony Vignaux (Institute of Statistics and Operations Research, Victoria University of Wellington, N.Z.) for the advise on simulator design and for providing his PasTime Pascal discrete-event simulatior construction routines, many ideas from which were incorporated in the Build module of AKAROA. Thanks also go to Mr. William Kennedy (Department of Electrical and Electronic Engineering, University of Canterbury), for supervising my work during the absence of my supervisor, and to Professor Penny (Department of Computer Science, University of Canterbury) for guidance with my written English.

The development of the AKAROA distributed simulation package was partially supported by a research contract with TELSTRA (Telecom Australia), and the development of SAM was partially supported by Telecom Coperation of New Zealand.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

There is growing interest in local lightwave networks because of the enormous bandwidth provided by cheap optical fibre, combined with the emergence of new bandwidth intensive applications. Twenty or even ten years ago bandwidth was the most scarce element of a communication system.

Early local networks used copper-wire or microwave-radio for interconnecting stations. Widely known first generation local networks include Xerox's Ethernet, IEEE 802.5 Token Ring, token bus (IEEE 802.3), and NSC's Hyperchannel. Copper based media have comparatively high attenuation (typically between one and 100 decibels per kilometre, depending on the type of media, the signal's fundamental frequency, and the operating environment) and is limited in bandwidth to typically 250 kilo Hertz (kHz) in the case of twisted pair, and 350 MHz for co-axial cable [STAL91].

More recently, fibre optic cabling has been commercially available, and has become cheaper per meter than copper co-axial cable. Optical fibre has its specific strengths in terms of very low attenuation (less than 1 decibel per kilometer within the low loss windows), immunity to electromagnetic interference, compact size, and a bandwidth of over 50 terahertz (THz) in the low loss regions of 1.3μm (about 25THz wide) and 1.55 μm (about 25THz wide). over four orders of magnitude greater than copper wire. As an illustration, a single strand of today's single-mode optical fibre has the bandwidth to carry a traffic equivalent to all the telephone traffic in the entire United States during the busiest transmission time (approx.10^{12} bps) [GOOD89].

Furthermore, in the region of 0.85 μm there is also a useful bandwidth of about 25THz. This region is of interest inspite of its higher attenuation (about 2dB per km), due to the fact that both lasers and their control electronics

that operate within this region can be made from gallium arsenide, a material already used in many commercially available integrated circuits (ICs).

The bit error probability of a fibre link is typically between 10^{-9} and 10^{-15} - almost ten orders of magnitude lower than that of copper twisted pair lines. The diameter of a typical commercial single or multimode fibre (including its cladding) is 125 microns (μ), just about twice the diameter of a human hair. Its small size permits high bundling densities.

Optical fibre is made from sand, one of the most abundant materials on Earth. The energy required to produce one kilometre of copper wire is sufficient to make over 900 kilometers of fibre [TOFF80]. As production volume increases and scale economies are achieved, the cost advantage of fibre over copper wire is likely to increase.

Second generation Local Area Networks (LANs) and Metropolitan Area Networks (MANs) use fibre as a substitute for copper to enable them to operate at slightly higher bit rates, typically at several tens of megabits per second. Examples include Fibre Distributed Data Interface (FDDI) and FDDI II [DAVI94], [REST94], [DYKE88], [KENN87] [JOHN87] and the Distributed Queue Dual Bus (DQDB) [MUKH92], [MART93], [CONT91], [IEEE90], [KIM90], [DRVA91] MANs. In these networks, each station has to have an electronic network interface that runs at the maximum network bit rate. The capacity of these networks is time-shared among their stations.

Second generation networks use fibre for its low attenuation and electromagnetic interference free properties. Nevertheless they can only use a tiny fraction of the bandwidth available in today's optical fibre. This is due to the fact that the maximum rate of the network equals the maximum transceiving rate supported by the electronic network interface of a station, which is limited to a few Gbps. So, in existing fibre optic networks, a medium which could support tens of Tera bits per second (one Tbps equals 1000 Gbps) capacity is throttled by electronic interfaces running at most up to a few tens of Gbps. The electronic interface of stations is therefore the main rate determinant, and is often referred to in literature as the *electronic bottleneck*.

With increasing complexity of existing network applications and the emergence of new uses, fast transport of higher volumes of traffic over a large area is required. High performance computing environments of the future are expected to depend on the network for interconnecting a variety of resources [CHLA90], [HENR89]. Components such as supercomputers and high performance workstations need interconnections and connections to high speed storage devices, as well as to color-graphics terminals or workstations where users can see full motion graphics. As an illustration, almost 800 Mbps of

effective bandwidth per user is required to pass uncompressed Super Video Graphics Array (SVGA) screen images of 1024 $\times$ 1024 pixels and 24 bits of color resolution at the typical TV frame rate (30 frames per second) from the host generating them to the remote display or window server.

Another high bandwidth application is distributed computing. Processing power comes cheaper in the form of multiple high performance workstations in a network than one fast supercomputer with their aggregate performance. But to employ distributed processing power on a job productively, computation has to be split into multiple processes for execution among the machines. Recently a number of mechanisms have been proposed and experimented with for distributed communication and synchronisation on a local network, including distributed semaphores, conditional critical regions [HANS72], monitors [HOAR74], message passing and remote procedure calls [BIRR84], some of which have been supported in the kernel level by several commercial operating systems. A major requirement for efficient interprocess communication is a fast network. As workstations and other intelligent devices get faster, more data is being transferred via the network. The degree of speedup obtained from parallel execution depends partly on the level of granularity of parallelism being exploited. Fine-grain parallelism potentially yields higher machine utilisation and better speedup, but involves more frequent interprocess communication, and therefore more network use. In addition to greater bandwidth use, low message transfer delay is often critical for capitalising on parallel processing. Distributed processes co-operating on one task are interdependent, so the progress of a subset of processes maybe blocked (idle), waiting for a message or response from others. In this situation, higher network latency could limit processor utilisation, thereby reducing speedup.

New applications include digital audio, high definition television (HDTV), video-conferencing and video-on-demand. All require high bandwidth and are delay sensitive. Medical imaging is another high bandwidth user, because it cannot accept lossy image-compression techniques [MUKH91].

With such future applications, it has been estimated that each future end-user will generate an average bandwidth demand of approx. 1 Gbps [GREE91]. With hundreds to thousands of users, the total network capacity would have to be in the order of Tera-bits per second.

Unfortunately it would be difficult to scale existing fibre based networks up to Terabit per second systems, because the maximum network rate that can be supported by the electronics at end stations is limited. Even if advances in electronic technology allow bit rates to be increased to 10-20 Gbps, only a minute fraction of the 30Tbps information carrying capacity of a strand of

optical fibre would be utilised.

One way of bypassing the electronic bottleneck is to use the network's electronic (bottleneck) resources in parallel. The throughput of conventional networks is limited to the maximum transmission rate of one station, e.g. on Ethernet or Token Ring. By enabling all stations to access the bandwidth of fiber in parallel, the aggregate electronic processing power of a network can be used concurrently, thus the total network bit rate equals that of a conventional network multiplied by the number of stations.

Parallel multiple access can be accomplished by multiplexing stations in either the spatial, temporal, or spectral domain, see Fig. 1.1.

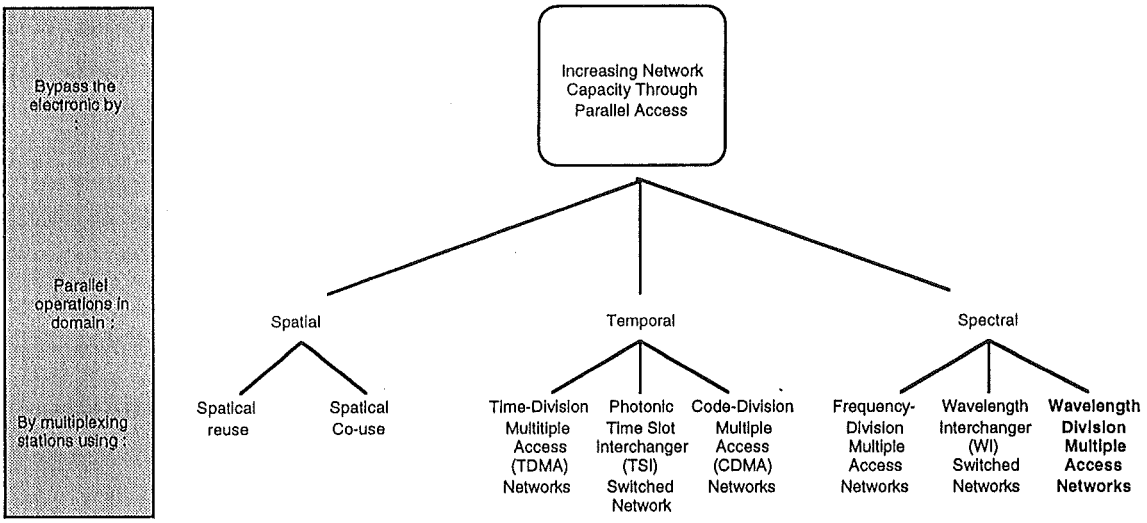


Figure 1.1: Bypassing the electronic bottleneck through parallel multiple access.

To allow stations simultaneous access to the same channel through spatial parallel access, the MAC procedure of the stations must discipline their operations so that their transmissions are confined to different spatial regions of the network at any one time. One application of spatial parallel access is *spatial co-use*, where the signals simultaneously transmitted by stations coexists on different spatial regions of the network. An example of spatial co-use in a conventional network is the DQDB network, where stations may transmit packets on different slots simultaneously. However, the speedup obtainable with this application of spatial co-use is limited by the fact that packets occupy channel space from their point of transmission, until they propagate to the end-of-bus terminator. This limitation, combined with the slot reservation procedure of

DQDB, restricts the average throughput rate per bus to less than one packet per time slot, even though the total packet transmission/reception rate that can be sustained by N stations is N packets per time slot per bus. Thus, although N simultaneous packet transmissions may be possible in certain instants, that rate cannot be sustained.

Another application of spatial co-use is the star network using an electronic space division switch [DEPR93]. All stations transmit on the same channel. Signals from different stations remain spatially separated from source-to-destination. Routing, space switching, and contention resolution is performed by the centrally located space division switch. This application achieves the ideal of parallel multiple access, supporting a sustained aggregate network throughput rate N times that achievable by one station, provided that the problems of destination conflict and switching speed can be solved. The main drawback with this approach is that the maximum network rate would be limited to the maximum rate supported by the electronic switch – an electronic bottleneck. Optical space division switch fabrics are currently an active area of research. Fabrics for switching space channels that are based on interconnections of either directional couplers [WA94], [LEE95], [CHIN95], or optical amplifiers, or arrays of light modulators [PENG96], [YAMA95], [MAO95], are under investigation. Currently, the set-up speed (time for establishing paths between input and outputs) of these fabrics is too slow for packet switching operations, due to the mechanism used for switching, or due to the speed of the electronics used to control the fabric. Also conflicts arise whenever two packets require switching to the same destination arrives simultaneously to the central switch. Only one of them could be switched to that destination's input, if a collision is to be avoided.

Another application of spatial co-use of the same channel is in Linear Lightwave Networks (LLNs) [STERN90], [KOVA95]. Here signals on a given channel may be confined to specific regions of the LLN during certain times, thereby creating the opportunity to co-use the same channel in different regions of the network. A greater degree of parallelism can therefore be supported using a small number of channels.

The other application of spatial parallel access is *spatial re-use*, where the signals transmitted by stations are erased upon arrival to a specific spatial region of the network, so that the bandwidth that they would otherwise have occupied can be re-used within the region. Applications of spatial re-use include the proposals for improving the DQDB network by allowing slot pre-use [PACH95], re-use [WEN94], and multi-use (which combines elements of slot pre-use and slot re-use). Slot pre-use allow slots on the channel to be

used and then erased (packet's signal absorbed) before they arrive to the station for which they are reserved under standard DQDB. Similarly, after serving a standard DQDB user, slots may be emptied and re-used. It has been shown that a channel utilisation greater than 100% can be obtained through channel re-use and pre-use. Obviously, the extent of increase in parallelism and hence performance improvement from spatial re-use in networks such as DQDB depends on the average minimum distance that a packet must occupy its slot. The distance that a packet must occupy equals the distance between the packet's source and destination.

For a given traffic pattern, the average source-to-destination distance of packets depends on the location of stations along the bus. The optimisation of station placement, so that spatial co-use and re-use could be maximised is therefore a rewarding problem when it is feasible to rearrange the placement of stations to match changing traffic patterns. Another interesting finding is that the benefits from spatial re-use in DQDB is not equitably shared amongst stations, so the characteristic of the unfair behaviour of standard DQDB is altered. This suggests the need for a modified Bandwidth Balancing Mechanism (BWB), if spatial re-use is to be adopted to improve the performance DQDB networks [YAU96c]. The advantage of spatial re-use is that the performance of DQDB could be upgraded without upgrading the speed of the transmission and reception that has to be supported by DQDB stations, and without increasing channel bandwidth. Another area of concern is that the degree of improvement, and the characteristic of the "unfairness" differs depending on whether slot pre-use or slot re-use or slot multi-use is employed.

The second approach to speedup using parallel multiple access is to operate in the time rather than the spatial domain. The bit (transmission/reception) rate sustainable by one user is only a fraction of the capacity of the fiber media. With *time parallel multiple access*, the information generated by a station is compressed in time, and share the media's bandwidth with many other stations in an interleaved manner. Thus the maximum network throughput can be N times the new packet generation rate of a single station. Note that either the per station bit duration is shortened, or a block of data (slowly) generated at the station is collected and compressed for transmission at a higher bit rate. In either case, stations can generate data for transmission in parallel, and the network throughput could be up to N times the generation rate of one station, where N is the number of stations in the network. Switching of the time multiplexed data can be achieved by an *active* network fabric, for example, using a photonic Time Slot Interchanger (TSI) [HUNT95], [KUWA94]. In this case, each station transmit and receive data during its own time-slot within each frame. Switching is achieved by interchanging the position in time of the

time slots in a frame of the time-multiplexed information stream, using the TSI.

Switching of the time multiplexed data can also be achieved by using a *passive* network fabric, for example a bus, ring, or passive broadcast-and-select-star. To illustrate, in a star network, the transmitter of a station may be fixed for transmitting during a specific time slot reserved for that station, within a frame. The star coupler combines the signals transmitted from all stations. The stations transmits during different time-slots of a frame, so the combined signals of stations would not overlap in time. The star coupler also broadcasts the combined signals to all stations. A station can therefore select which station to receive from by tuning its receiver for receiving information from any one of the time slots within a frame.

Another method for achieving parallel multiple access in the time domain is optical code division multiple access (CDMA). Many stations transmit simultaneously in the same frequency band, where stations use mutually orthogonal codes to represent bits. In the case of optical CDMA star networks, the pulse corresponding to a bit transmitted by each station is encoded using a set of delay lines (optical CDMA encoder) which scatters the pulse in time. To select the channel of that station, the destination sets its delay lines (tuneable CDMA decoder) to reassemble the (power) of the transmitted pulse. An advantage of optical CDMA is that new stations can be easily added to the network, as it only involves assigning the new station an code orthogonal to ones already in use.

A network can also support parallel station transmission/reception through parallel operations in the spectral domain. Like in time parallel multiple access networks, the switching of signals from source to destination station can be performed by the network's (active) fabric , for example, using a photonic wavelength interchanger (PWI) [FIJI88]. In this case, each station transmits on, and receives from a fixed wavelength (channel) dedicated to it. Thus stations need only fixed tuned single frequency transmitters and receivers. A connection is made between two stations by converting signals that are on the source station's wavelength to the destination station's wavelength. However, a problem occurs in PWI switched networks whenever two packets are destined for the same station, and simultaneously arrives to the PWI. If the PWI simultaneously converts all of them to the wavelength of their (common) destination, then all would be destroyed. Also the tuning speed and size of PWI may restrict its applicability to circuit switched applications where the connection holding time is long relative to the circuit set-up time.

Parallel multiple access in the spectral domain can also be achieved using

frequency division multiplexing (FDM). Each station in a network using FDM can simultaneously use several channels whose frequencies are electronically multiplexed together, and then the combined signal is used to modulate an optical carrier. FDM can therefore only partially alleviate the problem of the electronic bottleneck created by the limited electronic processing speed of stations. However, FDM can also be used in conjunction with wavelength division multiplexing (WDM). If each station is assigned one WDM channel, then FDM can be used by the station to use the optical bandwidth of the assigned WDM channel to support several lower bandwidth channels.

Another method for parallel multiple access in the spectral domain is wavelength division multiple access (WDMA), see Fig. 1.2. Many stations transmit simultaneously on different wavelengths. Stations then select the wavelength for receiving data using a filter which (ideally) passes only the desired wavelength. The procedure for switching a packet from its source to its destination depends mainly on the functionality supported by the network interface of the stations.

Each station is equipped with a network interface for transmitting and receiving data from the optical media. For this purpose, the major elements of each network interface will typically consist of a few single frequency laser sources (transmitters) and a few receivers.

A receiver works by firstly selecting the channel (wavelength) on which the station wishes to receive signals from, using an *optical filter*. The optical filter is a crucial component for the realisation of WDM networks. The optical signals on the selected wavelength are then received in a similar way to conventional fibre optic networks. That is, they are converted to electronic signals using a photodetector. Then the electronic signal is decoded using a conventional radio frequency receiver (RF-receiver).

The wavelength of lasers or filters may be fixed, that is, restricted to one particular wavelength (channel). Alternatively, they may be tuneable and can operate over a range of channels. In this case they are called agile (or tuneable) lasers and filters. A receiver whose filter is tuneable is called a *tuneable receiver*.

There are several physical topologies available for the optical interconnection network. The concern of this thesis is the class of WDMA networks based on a passive broadcast-and-select star topology, where each station is equipped with at most a few transmitters and receivers for data exchange. As an example, Fig. 1.3 depicts a broadcast-and-select star network with N stations. Each station is logically represented as two blocks: a transmission block, and a reception block. As shown, each station transmits signals on two

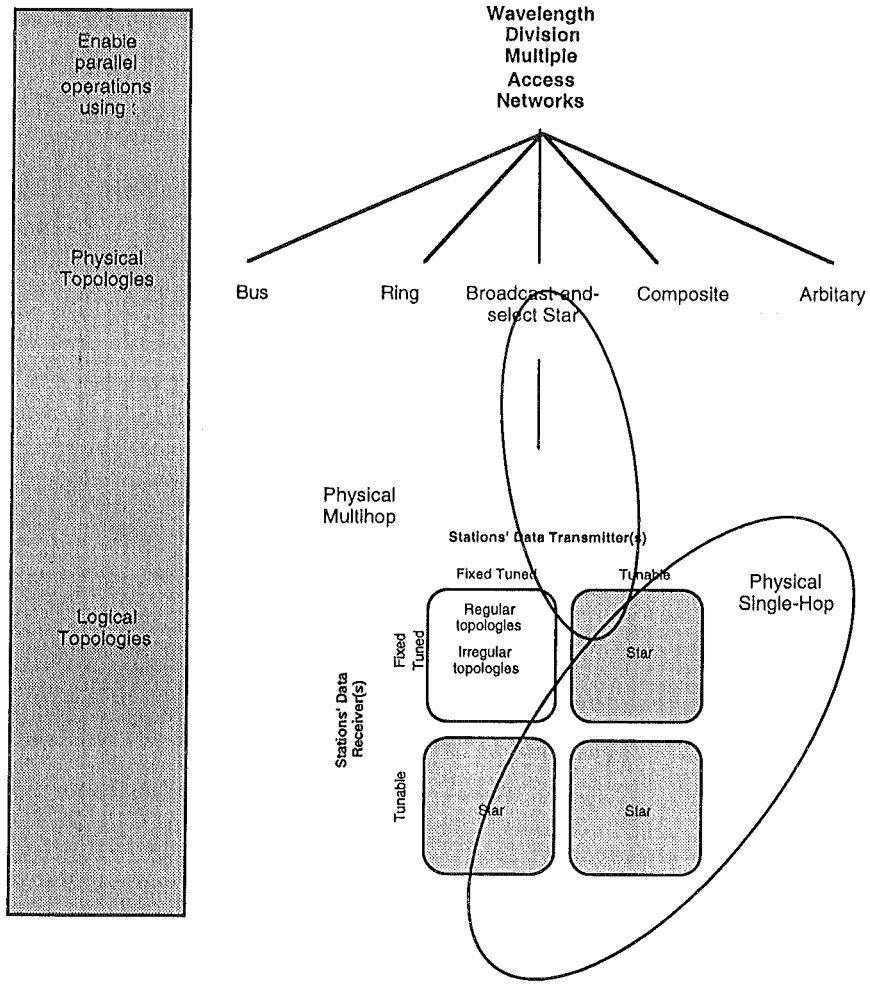


Figure 1.2: Parallel multiple access through wavelength division multiple access.

channels. Signals transmitted from stations are combined in a passive star coupler, and the aggregate optical signal is broadcasted from the star coupler to the receivers of all stations.

The star topology is favoured because of its more conservative power requirements if compared to the familiar bus and ring topologies [HENR89], [GREE93].

The diversity of network applications, spanning from low bandwidth constant bit rate consumers to bandwidth and delay sensitive variable bit rate applications, mean that the allocation of network resources among contending stations must be dynamically managed, for the network to be efficient. The emphasis of this thesis will therefore be on WDM star networks operating in a packet switching mode, with the understanding that circuit switched applications, or both, could also be supported.

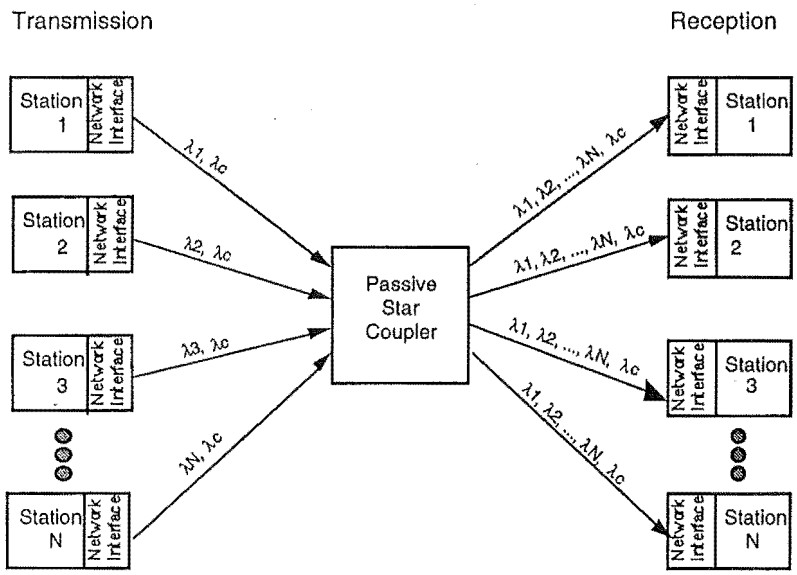


Figure 1.3: Broadcast-and-select WDM Star network

Early WDM star networks were designed using only fixed tuned transmitters and receivers. In these networks, called "multi-hop networks", each station is constrained to transmitting on and receiving from a subset of channels (see Fig. 1.2). Any-to-any connectivity is achieved by providing electronic switching within each node, together with a buffer for each transmitter and/or receiver. The connectivity pattern determined by the channels accessible by each node is called the *logical topology* of the network. Each packet is routed from node to node across the logical topology until it arrives at the destination node. Multi-hop networks carry low technology risk since they do not depend on the availability of tuneable transmitters or receivers. However, as the number of stations or the transmission rate increases, the dependence on electronic switching and routing of packets at intermediate nodes, and the need for each packet to traverse the physical network once per hop, may limit the feasibility and performance of multi-hop networks.

With advances in optical technologies, many WDM network designs were proposed assuming that fast tuneable optical filters would become feasible and cost effective. We focus on WDM star networks where each station uses a fixed tuned data packet transmitter (tuned to a unique data channel), and a tuneable receiver, for sending/receiving data packets, and fixed tuned transmitters/receivers for control purposes.

To exchange data, the origin station transmits a data packet on its dedicated channel. After one source-destination propagation delay, the destination tunes its data receiver to select the origin station's unique data channel, and receives the packet. Stations may simultaneously use their data channels without causing a packet collision. Consequently, the network interfaces of all stations of the network can be employed in parallel, and the total network bit rate equals that of a conventional network multiplied by the number of stations.

Section 1.1 describes the problem of destination conflicts arising from the parallel use of multiple channels that is addressed in this thesis. Previous solutions to the problem of destination conflicts in WDM stars are then briefly described in Section 1.2. The solution considered in this thesis, and the architectures and protocols for its implementation, and the method used for their analysis are outlined in Section 1.3. The organisation of this thesis is charted in the final section.

1.1 The Problem of Destination Conflict in WDM networks

One intuitively expects that higher throughput in WDM lightwave star networks can be achieved through the parallel use of multiple channels. However overall network performance may become less satisfactory, since the parallel use of multiple channels mean that more than one packet may be destined for the same station in the same time slot. If the station has only one data receiver, it can receive at most one packet per time slot. The other packets arriving for the same station would not be successfully received. This problem of *destination-conflict* leads to severe performance degradation [CHLA91], [CHEN91], [PAPA92].

Lost packets need to be retransmitted, limiting throughput to $1\text{-}e^{-1}$ (i.e. approximately 63%), [CHEN90], assuming independent and identical Bernoulli new packet generation processes at stations, and uniform distributions of packets' destinations. In fact, such a network is logically similar to an $N \times N$ switch where all but one of the packets simultaneously arriving for the same output are lost [KARO87][1]. It was shown [KAROL87] that the maximum throughput of such a system is also limited to $1\text{-}e^{-1}$ (approximate 63%), as $N \rightarrow \infty$.

[1]The difference being that packets in an N station network may be destined for one of N-1 stations, whereas packets arriving to an $N \times N$ switch may require switching to any one of the N outlets.

More critically, destination conflicts also lead to a larger average packet delay. Each retransmission of a lost packet would add one source-to-destination propagation period to its total delay, and propagation delay is expected to be high relative to the packet transmission time in high-speed networks. For example, if each station has one tuneable data receiver, then a station can receive just one packet during one time slot. A previous study showed that, assuming symmetric traffic and 40 stations, approximately 37% of the network capacity was lost to destination conflicts, and the mean packet delay can be up to 15 times the round trip propagation delay, due to the need to retransmit lost packets [CHEN91].

Stations must somehow detect destination conflicts, deduce whether their packets were received or lost, and then retransmit them if necessary. Resolving destination conflicts thus require extra electronic processing by stations. Large transmit buffers (at least with a capacity sufficient for storing also all packets that could be in transit) and more complex logic would be needed at the network interface of stations. Given that fibre is cheap and has abundant bandwidth, the station's network interface is a major cost determinant.

Consequently, effective resolution of the *destination-conflict* problem is critical if the potential opened by WDM is to be realised. Given the need for destination conflict resolution, a critical question is *where* within the network should this function be performed, and *when*. The choice of *placement* of this destination conflict function maybe critical, as it will have implications for the design of the network architecture and protocol, and hence the performance, computational complexity, and hardware demands of the network. However, the question of the placement of this function has not been explicitly considered yet. This thesis is concerned with choosing a placement so as to reduce the severity of the costs of resolving destination conflicts in WDM star networks.

1.2 Previous Work

As mentioned, the placement of the conflict resolution function comprises two aspects : the *location* where the function should be performed, and the time *when* it should be performed. In all of the previously proposed networks, this function is located at (performed by) at all user stations. One can organise previous networks into two classes depending on *when* destination conflicts are resolved :

1. Resolve *after* conflicts occur.

2. Resolve potential conflicts *prior* to packet transmission.

Many of the proposed networks assumed that channels are time-slotted. The duration of a time slot equals the transmission time of one fixed length packet plus the time for a receiver to tune from one channel to another.

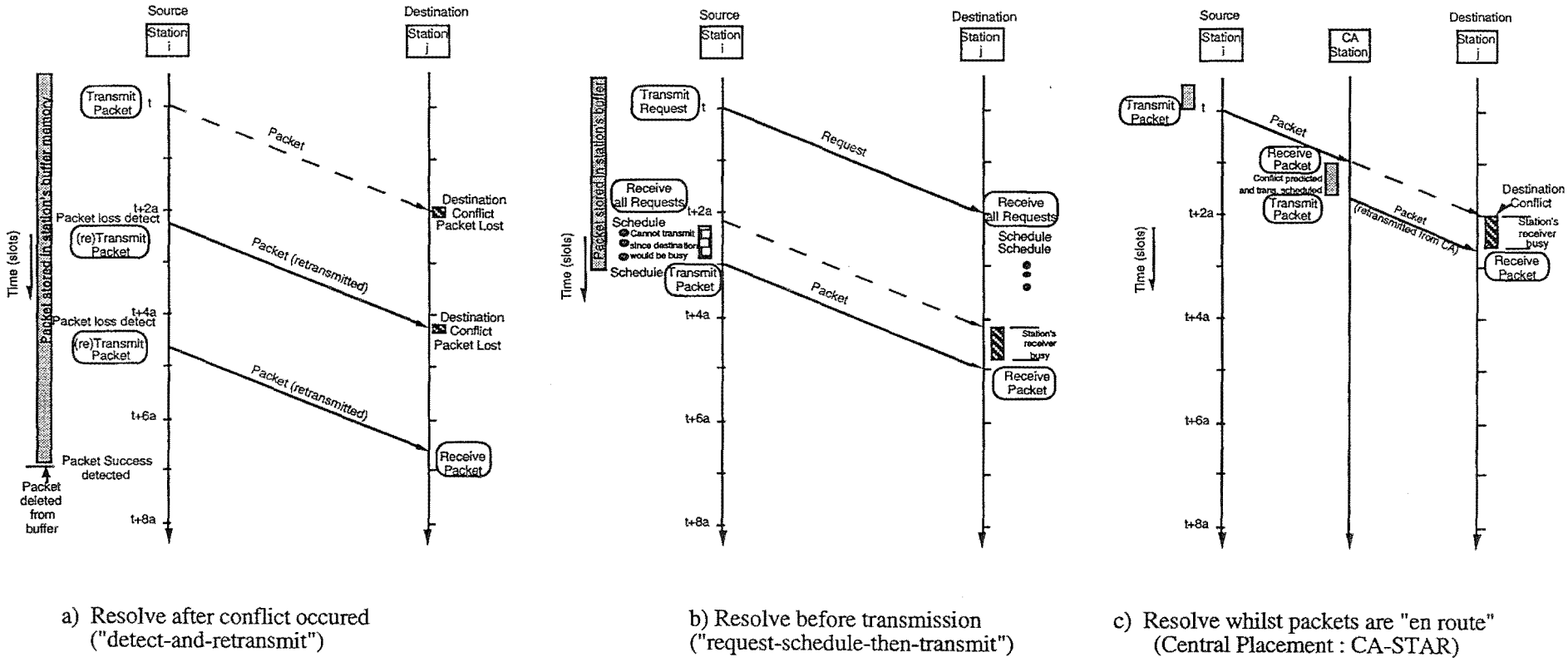
1.2.1 Resolve Conflicts *After* they Occur

In "detect-and-retransmit" networks, destination conflicts are resolved *after* they occurr. Stations transmit ready packets with at most one or two time slots delay, but packets may be lost due to destination conflicts. If lost, the source station must (somehow) detect failure and then retransmit the packet [CHEN90], [PAPA92], [GREE93], [HUMB93].

In [CHEN90] and [PAPA92] the source station determines the outcome of its packet transmissions from information it receives from a common control channel. In [HUMB93], one control and one data channel is dedicated to each station, so a station only has to process packet headers on its own control channel. Success is detected using acknowledgements and time-outs. Lost packets are retransmitted.

An interesting improvement to the "Detect-and-retransmit" method is to provide each station with d ($d \geq 1$) fast tuneable wavelength selective filters (such as the acousto-optic tuneable filter, AOTF [CHI95], [HINK93]) and d optical delay lines, for optically queuing packets so that each can be selected and switched for reception during (any) one of d contiguous slots. This significantly improves performance [CHLA94]. Origin stations are charged with detecting failure and retransmiting lost packets.

Network operation following the "detect-and-retransmit" principle is demonstrated in Fig. 1.4 by tracing the main steps for the transmission of a packet from station S_i to station S_j, assuming that the propagation delay between a station and the star coupler equals a time slots. Once the ready packet reached the front of S_i's transmission queue, S_i signals on a common control channel its intention to transmit it, then transmits the packet during the next slot. $2a$ time-slots later its outcome can be deduced by S_i based on information S_i receives from the control channel (If acknowledgements are used instead, the outcome is known after $4a$ time-slots). If the packet was lost, S_i retransmits the packet. This process is repeated until the packet succeeds.



a) Resolve after conflict occured ("detect-and-retransmit")

b) Resolve before transmission ("request-schedule-then-transmit")

c) Resolve whilst packets are "en route" (Central Placement : CA-STAR)

Figure 1.4. Approaches to resolving destination conflicts in WDM Star networks

1.2.2 Resolve Conflicts *Prior* to Packet Transmission

"Request-schedule-then-transmit" based networks [CHEN91], [CHIP93], [CHEN92], [FOOT95] resolves conflicts *prior* to packet transmission, see Fig. 1.4b. S_i firstly broadcast a transmission-request (on a time shared control channel) for its new ready packet, then waits.

The request is received by all stations $2a$ time-slots later, together with requests sent by other stations. Every station adds the requests received from the control channel to its Backlog matrix, during every time-slot. A Backlog matrix is a record of the destinations of all packets that are waiting at all stations.

Next, all stations execute the same destination-conflict-free transmission scheduling algorithm. The input to the scheduling algorithm is the information about packets that are waiting for transmission at all stations, as maintained in the station's backlog matrix. The scheduling algorithm uses this information for determining which packets waiting in stations' buffers could be transmitted during the next slot. The receiving of requests and the computation of a transmission schedule is performed by all stations during every time-slot. After its request had been delivered, the ready packet of S_i in question waits until a schedule allows its transmission. Once transmitted, successful reception is guaranteed.

Alternatively, *fixed transmission schedules* have been studied, where source-destination transmission rights are predetermined for all data channels and slots [GANZ89], [CHLA87], [CHLA88], [CHLA90], [ROUS93]. The transmission rights schedule can be designed to avoid destination conflicts. Obviously, in such a case, packets have to be buffered at their source stations awaiting (fixed) times when they can be transmitted, even if the network is idle in the meantime. Excellent network utilisation can be achieved, but this approach may not be well suited for stochastic traffic, or under light load. Under these conditions, fixed assignment schemes would lead to a very inefficient use of the available optical bandwidth [MEHR90], and unnecessarily high packet delay.

1.2.3 Receiver Replication

An alternative approach to the problem of destination conflicts is to equip each station with N receivers if N data channels are used by the network. Each station thus can receive from all data channels simultaneously [SENI91].

1.2.4 The Underlying Cause of Conflicts in WDM networks

The fundamental dilemma behind the problem of destination conflicts is that stations are *geographically separated*. At any instant, a station could not know the status nor transmission intentions of other stations.

If stations transmit their packets without knowing the actions of other stations, destination conflicts can occur. This happens in "detect-and-retransmit" networks, where destination conflicts are detected, and resolved *after* they occur.

On the other hand, a station could attempt to ascertain the status of all other stations (i.e. the intended destinations of packets waiting for transmission), and use this knowledge to schedule its transmissions so that destination conflicts are resolved *prior* to packet transmission. This might mean high electronic processing overheads. But the major drawback is that each packet is delayed for at least one source-to-destination propagation period *prior to transmission*. If the propagation delay is high compared to the transmission time of a packet, this may be undesirable.

1.3 Resolving Conflicts Whilst Packets are *En Route* to their Destinations

This thesis considers a *central placement* of the destination conflict resolution function whereby *only one central station* located at the entrance to the star coupler is responsible for detecting conflicts and re-scheduling the arrival times of "otherwise lost" packets whilst they are *en route* to their destinations, so that the packets arrive to their destinations when they are free to receive them.

1.3.1 Centrally Arbitrated WDM Star Networks

This thesis considers a different approach to the same destination-conflict problem, whereby stations may transmit packets without waiting for a long contention resolution period, and assume that each of them is successfully delivered. Thus, the minimum packet delay prior to transmission is just two slots, independent of the propagation delay. Since transmission proceeds without firstly ascertaining the transmission intentions of other stations, more than

one packet may simultaneously be destined for the same destination, but the destination will be able to receive only one of them. A conflict arbiter (CA) located at the entrance to the star coupler detects destination conflicts, and re-schedules the arrival times of packets which would otherwise be lost[2] (thereby rescuing them whilst they are *en route* to their destinations), so that they reach their destinations when their destinations are free to receive them, see Fig. 1.4.

Consequently, even if packets are involved in a destination conflict, they would be delayed only until their destinations are free. Both electronic memory and optical delay lines are considered for storing otherwise lost packets at CA, until their re-scheduled times of departure from CA. If optical buffers are used, then once transmitted a packet remains in the optical domain *even if a destination conflict occurs*. This "central placement" approach may employ the same proven ideas of conflict-free scheduling [CHEN91], [CHIP93], [CHEN92], [CHEN94], [FOOR95], [YAU96b] and delay lines [CHLA91], [CHLA94], but re-positions them in space (the conflict resolution function is performed at the entrance to the star coupler) and time (performed whilst packets are already en route to their destinations) to try to obtain performance, complexity, and cost advantages. Let the network based on this *central placement* of the conflict resolution function be called a CA-STAR network. In deference to their distinguishing trait of resolving conflicts whilst packets are en route to their destinations, the central placement of the conflict resolution function may also be referred to hereafter as "en route" conflict resolution.

Unlike a centralised electronic switch or a station in a multihop network, CA does not perform switching nor routing functions. As mentioned, CA can be designed for optical or electronic implementation. Electronic buffering of is also attractive, since a packet would require rescuing (buffering) by CA only if it needs to wait one or more time-slots before its destination is free to receive it. The delay of one time-slot for optical-to-electronic conversion of "otherwise-lost" packets is thus *desirable delay*. CA can be designed so that its buffer operations proceeds at the same speed as that of ordinary stations, so they would not create an electronic bottleneck.

Network operation following the "en route" conflict resolution principle of CA-STAR is demonstrated in Fig. 1.4c. S_i firstly signals on the common control channel its intention to transmit the ready packet, then immediately transmits the packet itself (without waiting until the signal has been deliv-

[2]When more than one packet simultaneously arrive for the same destination, the destination can receive only one of them. The other packets can therefore be considered as "otherwise lost" packets which needs to be rescued by CA.

ered). The outcome of the packet transmission can be deduced by CA prior to its arrival to CA, based on information CA receives from the control channel. If the destination can receive the packet, it would remain in the optical domain until it is received (dashed arrow). If the packet is expected to be lost (its destination will not be free to receive it during its original arrival time), it must wait until its destination is free to receive it (at least one time-slot). CA receives the packet (one slot desirable delay), and then delays its retransmission until the first slot when its destination would be free to receive it (desirable delay). Packets transmitted from CA always succeed.

The relative efficiency of the various network architectures depends on the ratio between station-to-star-coupler propagation delay and packet transmission time (a).

$$a = \frac{station-to-star-coupler\ propagation delay}{packet\ transmission\ time}.$$

When $a < 1$, the delay of a packet is dominated by the electronic-to-optical (E/O) conversion time for the packet transmission(s), and for O/E conversion for its reception(s), plus the transmission/reception times of requests or control signals.

"Detect-and-retransmit" based networks need a minimum of 2 conversions per packet, plus the time for success detection. Every time a packet is lost, a copy of it has to be retransmitted (E/O conversion) and its success/failure monitored (O/E conversion of control signals).

CA-STAR networks also need a minimum of 2 conversions per packet plus the time for success prediction by CA. The main difference is that if a packet is expected to be lost, it would experience only a delay of one time slot for its reception by CA (assuming electronic buffering is used). This delay of rescued packets is *desirable* because their destinations would not be free to receive them until the later time slots. Once a rescued packet is retransmitted from CA, it would be successfully received.

Nevertheless, a is inversely proportional to transmission speed, and directly proportional to network diameter. In optical MAN and LAN applications, high transmission rates (hundreds of Mbits/sec. or even several Gbits/sec. per channel) and distances up to tens of kilometers are expected. In these environments $a > 1$, and it is expected that significant performance improvements in terms of throughput and packet delay can be obtained with *en route* conflict resolution, as demonstrated in Fig. 1.4. This is due to the fact that in "detect-and-retransmit" networks, a packet experiences added propagation delay of $2a+1$ time-slots each time it is lost, plus the one source-destination propagation period. In CA-STAR even if a packet is involved in a destination

conflict, its delay due to propagation still equals one source-destination period. Moreover, multiple retransmissions are unnecessary, so significant throughput improvements are also expected.

An advantage of CA-STAR over "request-schedule-then-transmit" networks is the resolution of destination conflicts without having to wait for a long request-and-schedule phase. In a CA-STAR network, new packets are transmitted almost as soon as they are generated. Only packets that would otherwise be lost are "rescued" by CA, and even so they would be delayed at CA only until their destination is free to receive them (Fig. 1.4c). In "request-schedule-then-transmit" networks, a packet must wait until it is at the head of its transmission queue, and then wait at least $2a+1$ time-slots during its request broadcast, and then wait until a schedule permits its transmission (Fig. 1.4b). CA-STAR is thus expected to yield a reduction in mean packet delay of at least $2a+1$ time-slots. Furthermore, CA-STAR is expected to offer improved throughput when $a > 1$, since the information used for scheduling in "request-schedule-then-transmit" networks would be $2a+1$ slots outdated.

Moreover, with CA-STAR, only the CA station needs to resolve destination conflicts. CA is sited at the entrance to the star coupler, so data signals from stations propagating through CA are still Space Division Multiplexed (SDM). Consequently, *CA needs to rescue at most one packet per station per time slot* (see Fig. 1.4c). Thus CA does not require multiple filters nor receivers per station. Due to SDM inputs, neither optical filters nor tuneable devices are needed for conflict resolution.

Dividing the cost of CA amongst stations, the cost per station of a CA-STAR network can compare very favourably with alternative architectures. Amongst the benefits introduced with CA-STAR, it is important to mention the cost reductions from lower buffer memory use, and complexity inversion. The light shaded bars along the time axis of Fig. 1.4a 1.4b and 1.4c indicates the duration when a packet needs to be stored in electronic memory within the network. In CA-STAR, a packet is transmitted almost as soon as it has been generated, and can immediately be deleted from the station's buffer. Packets are buffered by CA only if necessary. Thus the memory requirement of the CA-STAR network is less than that of "request-schedule-then-transmit" and "detect-and-retransmit" networks, where each packet is buffered for at least $2a+1$ time-slots. Also the buffer capacity of CA-STAR does not have to be upgraded as a increases.

Tasks such as success detection, retransmission, or schedule computation are replicated by all stations in "request-schedule-then-transmit", and "detect-and-retransmit" networks. In CA-STAR, they are performed by just the CA

station. The processing complexity of ordinary stations can therefore be drastically reduced.

1.4 Structure of this Thesis

The organisation of this thesis is summarised in Figs. 1.5 and 1.6.

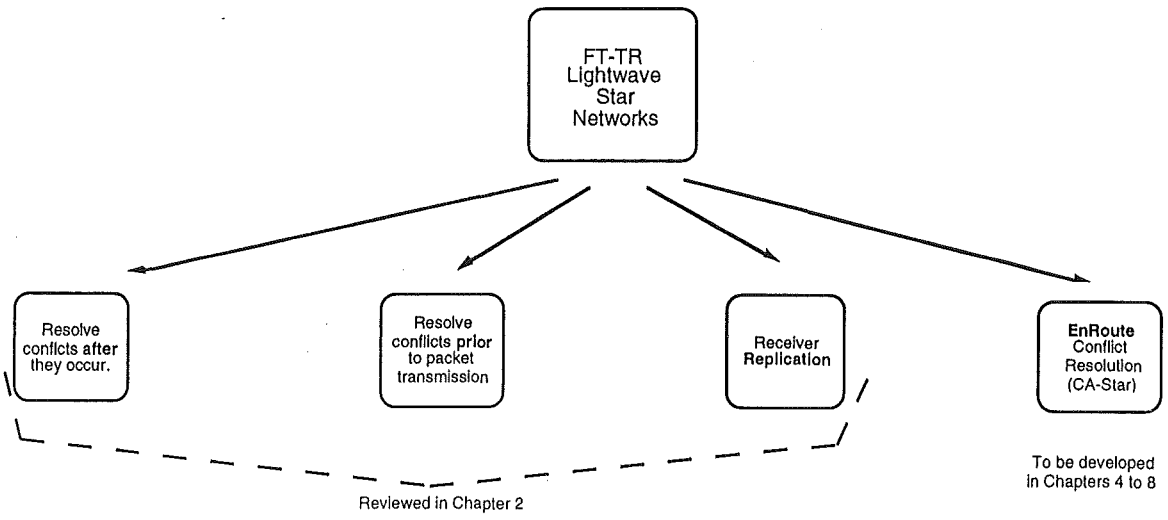


Figure 1.5: Approaches to resolving destination conflicts, and their order of presentation.

1.4.1 CA-STAR Architectures

Three central arbiter designs are considered in this thesis, see Fig. 1.6. The first central arbiter assumes the use of shared memory for temporarily storing rescued packets. It is named sCA, and the CA-STAR networks implemented using sCA are named sCA-STAR networks. The second central arbiter was designed for either **opt**ical or electronic buffering of "otherwise lost" packets, assuming the availability of wavelength converters. Therefore it is named **opt**CA, and CA-STAR networks implemented using optCA are named optCA-STAR networks. The third possible realisation of the central arbiter was

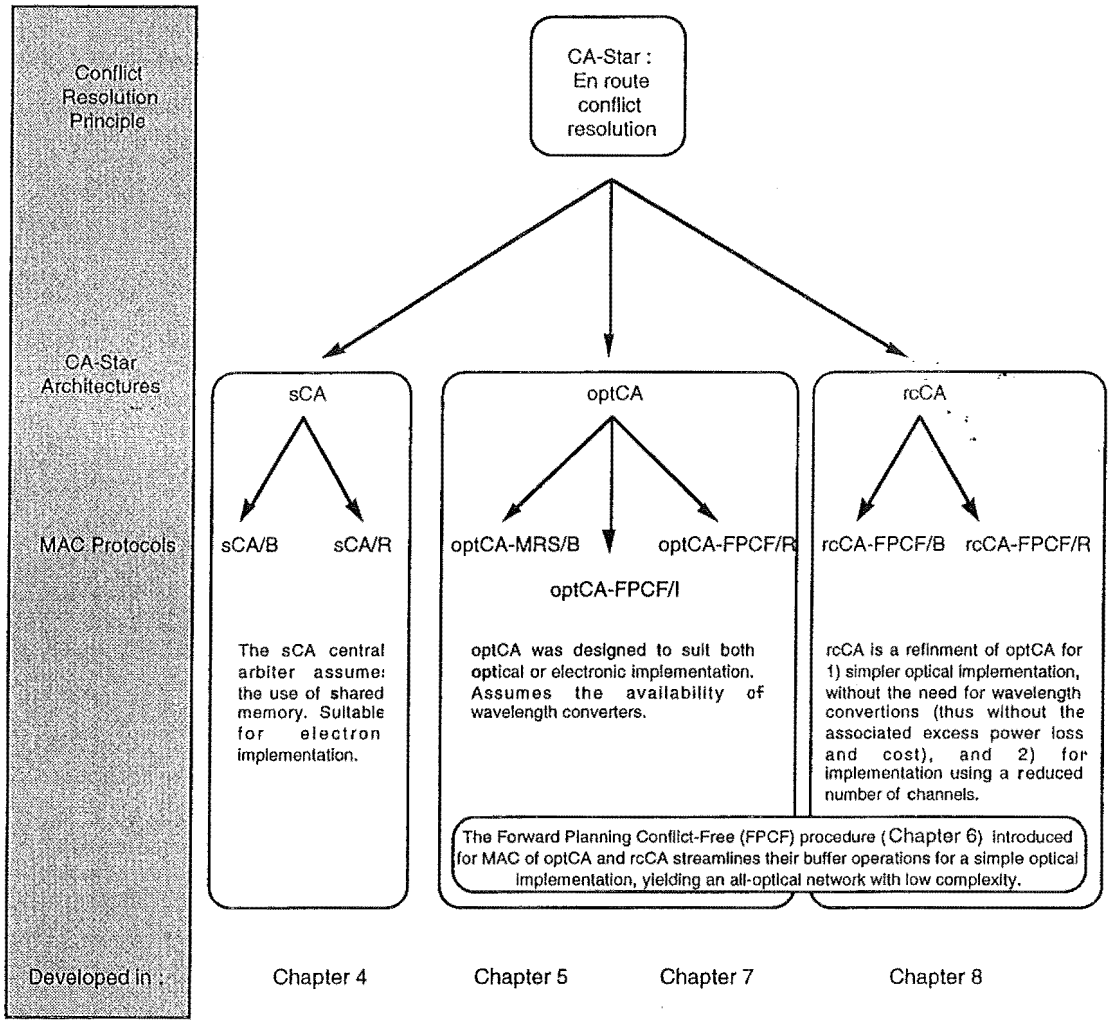


Figure 1.6: CA-Star architectures and protocols considered in this thesis.

designed for simple optical implementation without the need for wavelength converters, to give an all-optical network of low complexity that provides en route conflict resolution. This design also has reduced channel requirements (i.e. can be built using less channels than previous networks). This central arbiter is named rcCA, and CA-STAR networks implemented using rcCA are named rcCA-STAR networks.

1.4.2 Protocols

Two types of media access control (MAC) protocols are proposed for each CA-STAR architecture. The first type, called type B (since it has a Bounded delay property), is intended for network applications where the major bandwidth consumers are delay sensitive services which can accept some packet loss, provided that the probability of packet loss is below a specified level. The

second type is called type R, since its use a mechanism called reflection for preventing any losses of packets due to temporary depletion of buffer resources. Both types are described in greater detail below.

The type B protocols are characterised by having three properties :

1. **B**ounded packet delay. The maximum delay of a packet is bounded.
2. Packets may be lost due to destination conflicts, but the probability of packet loss can be kept below a given maximum value, by using a connection acceptance control function.
3. The protocol preserves the ordering of packets from a given source station that are destined for a given destination station.

Let us identify protocols of this type by including the suffix B (for Bounded delay) in their names. The protocols of this class are shown in relation to the CA-STAR architectures in Fig. 1.6.

This type of protocol was designed for network applications where some packet loss is acceptable, provided that the probability of packet loss is below a specified level. Once a station transmitted a packet, it assumes that it will be successfully delivered, even though there is a probability that it could be lost. The probability that a packet is lost depends on the pattern of arrivals to its destination station, as specified by the probability density function of the packet arrival process to that destination. Thus, in connection oriented-dynamic bandwidth allocation packet switched networks, the probability of packet loss can be kept below a specified level by the use of a connection acceptance function by stations when deciding whether to accept new connection requests. These protocols may suit applications where the major bandwidth consumers such as HDTV, video-on-demand, voice, and teleconferencing applications can tolerate some packet loss. Loss averse users such as distributed databases can also be accommodated, if the probability of packet loss is not too high. For these applications, the losses of packets must be detected by higher protocol layers and the protocol data unit (PDU) containing the lost packet retransmitted.

In CA-STAR networks, CA is tasked with rescuing "otherwise lost" packets. However, CA has finite buffer capacity, so the question is raised of what to do with packets that need to be buffered by CA, when CA's buffer memory is full. If we assume that once a station transmits a packet, its successful delivery can be assumed, then there is no need for acknowledgements nor success detection, and there is no need for the station to keep a copy of the transmitted packet in its transmit buffer until the packet has been received.

The type R protocols are targeted for network applications where packets must *never* be lost due to buffer overflow at CA. One possible method for eliminating packet loss due to buffer overflow in CA is proposed in this thesis. This technique is called *reflection*. We shall name protocols that use reflection by including an /R in their names. The protocols of this class are shown in relation to the CA-STAR architectures in Fig. 1.6.

Convention for Naming Protocols

A protocol for a CA-STAR network can identified according to

1. The CA-Star architecture that it serves. A protocol can be designed for the operation of either the sCA-STAR, optCA-STAR, or rcCA-STAR networks.

2. The algorithm used by the MAC procedure of CA for resolving destination conflicts. The choice of algorithm determines the computational complexity of the MAC procedure, and the performance of the network.

3. Its *type*.

The names of CA-STAR protocols are formed by concatenating 1) the name of the central arbiter on which its intended architecture is based (either sCA, optCA, or rcCA), 2) The name of the conflict resolution algorithm employed by the CA station (if any), and 3) the suffix identifying its type (either /B or /R). For example, the optCA-MRS/B protocol refers to the protocol for a CA-STAR network implemented using the optCA central arbiter, where the MAC procedure of optCA uses the MRS algorithm for conflict resolution, with the above properties of type /B protocols.

The CA-STAR architectures and protocols considered in this thesis, and the chapters where they will be introduced, are summarised in Fig. 1.6.

As shown, several protocols will be considered for each CA-Star architecture. A CA-STAR network operating according to a specific protocol will be referred to simply by its protocol name. For instance, by "an optCA-MRS/B network" we refer to a network with the optCA-STAR architecture operating according to the optCA-MRS/B protocol (just as "a DQDB network" refers to a network of the DQDB architecture, operating according the DQDB protocol).

1.4.3 Analysis Methodology

A simulation tool is needed for the performance analysis of CA-STAR architectures, each operating according to one of several media access control protocols, under various modelling assumptions, over a multi-dimensional design region, and for obtaining results of specific accuracy within a practical time frame. These considerations motivated the development of a new parallel/distributed simulation technique called Spectral Analysis in Parallel Time Streams (SA-PTS).

SA-PTS has been implemented in an object-oriented automated parallel simulation package called AKAROA. It was assumed that AKAROA should accept ordinary (non-parallel) simulation programs, and all further stages of stochastic simulation should be transparent for users. Such a package should automatically transform sequential simulators into ones suitable for parallel execution. At runtime, simulations based on SA-PTS exist as a set of co-operating parallel processes, possibly executing on several machines interconnected by a local area network. Thus multiple simulation processes need to be created at runtime. When they need to contact a co-operating process, the machine address and port number of that process must be (somehow) located, or one must be created if it does not yet exist. These tasks, and the control of the precision of estimates, and stopping of all parallel replications and global precision control processes when the required precision of all steady-state estimates have been achieved, should be performed transparently.

AKAROA consists of two main modules: Parallel Simulation Manager, responsible for the automatic creation of simulation and global precision control processes, process management, and interprocess communication; and Control, responsible for controlling simulation run-time and analysis of output data collected during steady-state simulation. The package is also equipped with Build, a module which can be used for rapid construction of typical simulation models. Users have access to services offered by AKAROA through a simple programming interface. At run-time the set of Parallel Simulation Manager processes of AKAROA cooperates with the user runtime interface process to present the multiprocessor and/or network of workstations as one (virtual) uni-processor to the user.

The design issues, architecture, and implementation of AKAROA, as well as the results of its preliminary performance studies are presented in a separate technical report [YAU96a]. The main performance indices evaluated were *coverage* (defined as the percentage of confidence intervals of estimates that contain the true parameter), *run time speedup* (defined as the ratio of the

execution time on one processor to the execution time when P processors are employed), and *inter-processor communication overhead* (represented by the number of datagrams exchanged between processes participating in simulation execution).

1.4.4 Main Contributions

The main contributions of this thesis lie in the consideration of en route conflict resolution in lightwave star networks. Also, architectures and protocols are studied for the realisation of en route conflict resolution, and their performances were analysed using distributed stochastic steady state simulation.

1.4.5 Organisation

Chapter 2 discusses in greater detail the WDM architectures and protocols proposed in the literature.

Chapter 3 diverts our attention to considering the main hardware elements of the class of WDM networks of concern : namely the single frequency laser, tuneable filter, and the star coupler. The main assumptions on their characteristics required by the networks are given, followed by a review of the principle of operation and the characteristics of currently achievable devices.

Chapter 4 presents CA-STAR architectures and protocols based on a central arbiter that uses shared memory (sCA), and the analysis of sCA based CA-STAR networks (referred to as sCA-STAR networks).

Chapter 5 addresses the problem of memory access speed mismatch between the operations of the transmit buffer of ordinary stations and the buffer of sCA by introducing the optCA central arbiter. The implementation of optCA only requires a buffer access speed equal to that of the access speed of the transmit buffer of ordinary stations. Speed parity mean that CA's memory operations would not develop into the system's bottleneck, as the transmission rate of ordinary stations increases. The optCA design is suitable both for electronic and optical implementation.

Chapter 6 presents a new traffic assignment algorithm for selecting packets in the buffers of optCA for transmission in a conflict-free way. The proposed algorithm differs from previously introduced algorithms for the above tasks in that it employs *Forward Planning* when making transmission/reception decisions. The algorithm is analysed by first evaluating its throughput and delay

characteristics, comparing them with that of the SDR algorithm that features 100% assignment efficiency. Then the worst case computational complexity of the proposed algorithm is evaluated and compared with that of SDR, and several suboptimal but low complexity algorithms reported in the literature. Since the need for conflict free traffic assignment may also arise in systems where packets (requests) at input buffers (processors) are directed to specific outlets (modules), the results of this study may have wider implications.

In Chapter 7, new opt-CA-STAR network protocols based on the proposed traffic assignment algorithm are defined, followed by their analysis.

Chapter 8 addresses the problem of achieving optimal channel utilisation, introducing an architecture that can be built using a reduced number of channels (rcCA-STAR).

The CA station of an rcCA-STAR network stores (buffers) rescued packets in a logical pipe with constant flow (emptying) rate. An optical delay line (i.e. a "loop of fibre") naturally serves as a delay pipe. Also, at the rcCA, the optical signals from stations are still space division multiplexed. Thus the optical delay line can be implemented without using multiple fast tuning optical filters, nor wavelength converters, nor any tuneable devices. The implementation of rcCA-STAR using optical buffering is considered in the second section of Chapter 8.

Lastly, in Chapter 9, the performance, MAC protocol computational complexity, buffer organisational complexity, hardware demand, and fault tolerance of CA-STAR networks (based on "en route" conflict resolution) are compared with each other, and with networks that are based on the "detect-and-retransmit", "request-schedule-then-transmit", "detect-and-retransmit enhanced with multiple delay line and wavelength selective switches", and "receiver replication" approaches to solving the problem of destination conflicts.

The final chapter focuses on some implications.

Chapter 2

Previous Work

This chapter provides the background for locating our interest within the area of broadcast-and-select star WDM networks. A classification of WDM networks is proposed, together with the main MAC problems encountered by each network class. Previous WDM systems within each class are reviewed, with emphasis on networks belonging to the same class as CA-STAR.

The media access problems confronting a specific WDM star network architecture depend mainly on the functionality supported by the network interface of its stations. We have therefore grouped previously proposed architectures according to whether the *data* transmitter/receivers of their stations are rapidly tunable or fixed tuned. Stations in each class may also use additional transmitter(s) and/or receiver(s) for solving media access control problems. The classification of broadcast-and-select star WDM networks are summarised in Fig. 2.1.

In the first class of WDM networks, each station is provided with m fixed tuned optical transmitters (FTs) and m fixed tuned receivers (FRs) for data transfer (FT-FR class). Each transmitter in the network is fixed tuned to a different wavelength. In general $m < M$ where M is the number of channels, so a station could only receive from a subset of channels. Any-to-any connectivity between stations is achieved using multi-hopping, whereby a packet is routed from node to node until it arrives at the intended destination [ACAM91], [ACAM92a], [ACAM92], [ACAM94], [BANE91], [BANE94], [BANN90a], [BANN90], [CHAN93], [ELBY94], [FORG93], [FORG95], [FRAT94], [GANZ93], [INES95], [KARO91], [KOVA94a], [KOVA94], [LABO91], [MAXE87], [MUKH92], [TANG94a], [TANG94], [WILL93], [ZHAN91], [ZHAN94]. In such *multi-hop* networks, the problems of wavelength assignment, routing, buffer dimensioning, and admission and congestion control may need to be solved.

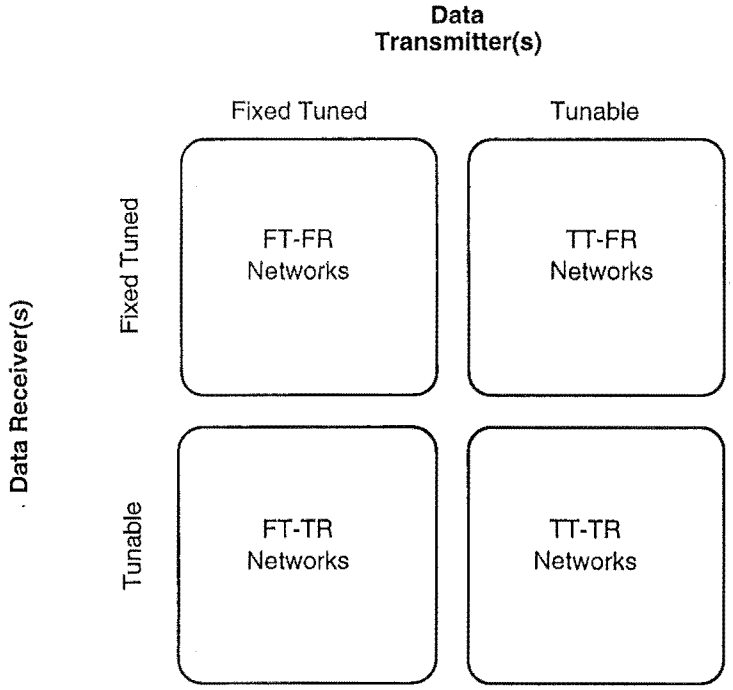


Figure 2.1: Classifying WDM Star Networks according to the functionality provided by the transceivers of their stations

In the second network class, each station is equipped with one tunable transmitter and one fixed tuned receiver for data exchange (TT-FR class). *Collisions* occur if two or more stations transmitted packets on the same wavelength and they arrive to the hub (star coupler) at the same time. All packets involved in a collision would be destroyed[1].

In the third network class, each station is fitted with a transmitter, for sending data at a fixed wavelength unique to that station, and a tunable receiver for data reception (FT-TR class). A *destination conflict problem* may arise since the tunable receiver can only tune to and receive from a single wavelength at any given time. If more than one packet arrives for the same station during one time slot, the station is able to receive only one of them. Moreover, a mechanism must be provided for informing destination stations in advance of which wavelength(s) on which there are packet(s) destined for them. This is called the *transmitter-receiver co-ordination* problem. CA-STAR belongs to the FT-TR class.

[1]We are not aware of any proposed WDM network where the "capture" effect has been used to alleviate the problem of collisions, and in general the use of different power levels by different stations may be impractical given the limited power budget of transmitters (power bottleneck, see Chapter 3).

Lastly, in networks of the TT-TR class, stations use a tunable transmitter for data transmission, and a tunable receiver for reception. Here, the major problems are packet collisions, and transmitter-receiver co-ordination.

Fig. 2.2 summarises the *major* MAC obstacles in each network class that have to be overcome in order to realise the potential unlocked by WDM.

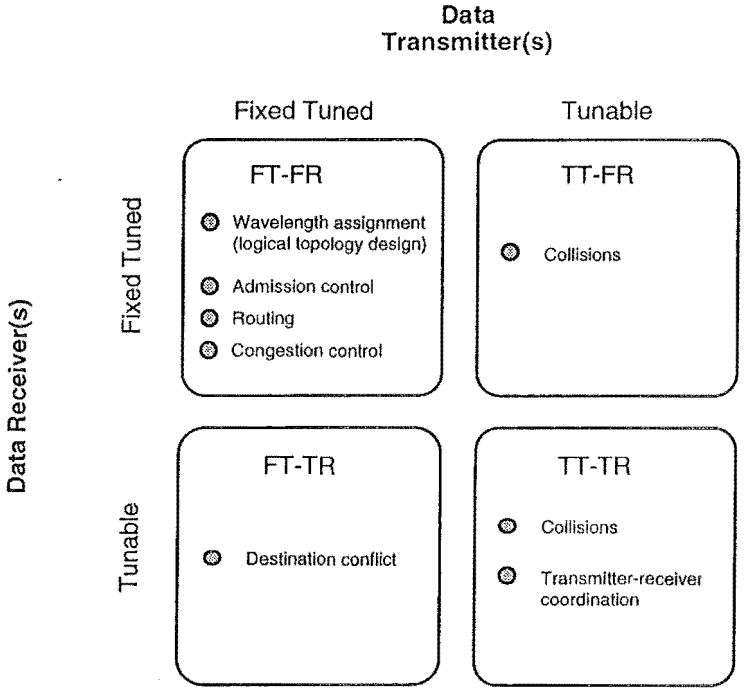


Figure 2.2: Major obstacles to the realisation of potential network performance faced by each network class.

Previous network architectures and protocols within the FT-FR, TT-FR, and TT-TR classes are described in Appendix A, B, and C respecitvely. The networks considered in this thesis belong to the FT-TR class of WDM star networks. Thus the rest of this chapter is devoted to reviewing previously proposed FT-TR networks.

2.1 FT-TR Networks: Review of Problems and Solutions

In FT-TR networks, each station is equipped with a transmitter, for transmitting data at a fixed frequency unique to that station, and a tunable receiver for data reception.

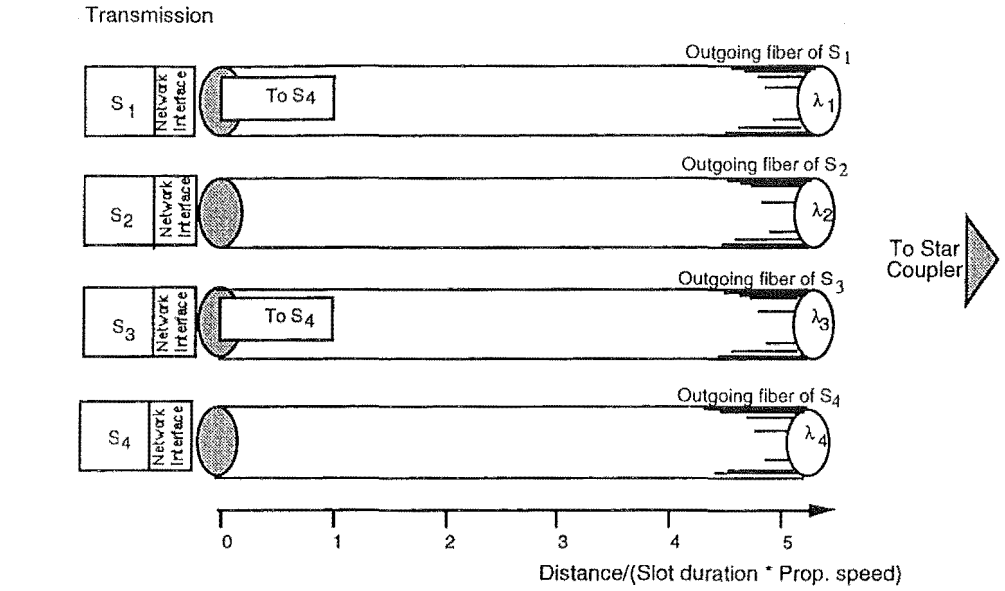
Stations communicate in the following way. Each station has its own data channel for transmitting its data packets. A busy station transmits its packet on its data channel. After one source-to-destination propagation delay, the destination station tunes its receiver to the data channel of the source station to receive the packet. The exception to this procedure is when a destination conflict occurs. A *destination conflict problem* may arise since the tunable receiver can only tune to, and receive from, one channel at any given time.

The merit of FT-TR networks is that it permits a packet to be exchanged between any pair of stations within the optical domain. That is, once transmitted, a packet would remain in the form of light until it is received by its destination – *unless it is involved in a destination conflict* with another packet. FT-TR networks also have the potential to support broadcasts/multicasts in a natural way. Like TT-FR networks, an advantage of FT-TR networks compared to networks of the TT-TR class is that FT-TR networks demand just one tunable component (the TR) per station [BOGI93b]. The drawback of FT-TR networks is the problem destination conflicts.

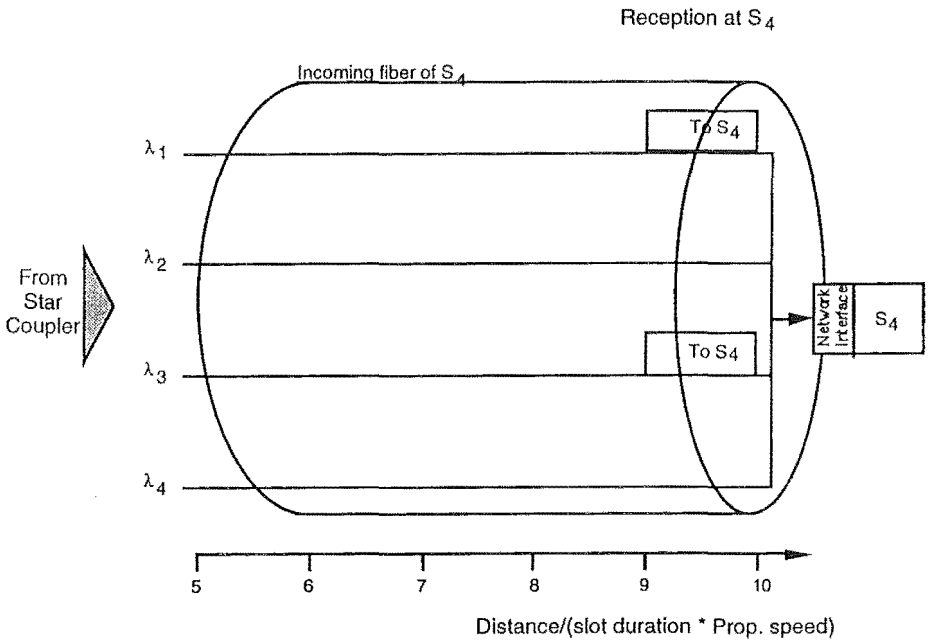
2.2 The Destination Conflict Problem

Fig. 2.3 illustrates the *destination conflict* problem in FT-TR systems. A 4 station network is depicted, where all stations are a=5 slots from the star coupler. During time-slot t station S_1 transmits a packet to station S_4 on its outgoing channel (λ_1), and station S_3 transmits a packet to S_4 using its own outgoing channel (λ_3). After two source-to-hub propagation delay periods, i.e. at the start of $t+2a$, both packets arrive at the receiver ports of all (4) stations.

The problem of destination conflict in this case is: two channels, λ_1 and λ_3, contain packets destined for station S_4 during $t+2a$. Having one tuneable data receiver implies that a station can receive at most one data packet during a slot. Hence S_4 faces a problem: it can receive only one of the two packets destined for it, loosing the other.



a) Transmission during time slot t



b) Reception at S_4 during time slot $t+10$

Figure 2.3: The Destination Conflict problem in FT-TR networks

The problem of destination conflicts necessitates the implementation of a destination-conflict-resolution function somewhere within the network, and a major design decision is the placement of the destination-conflict-resolution function which specifies the location(s) *where* it should be performed, and *when* it should be performed. In all of the previously proposed networks, this function is located (performed) at all user stations. Therefore one can classify previous networks according to on *when* destination conflicts are resolved, see Fig. 2.4.

- Resolve conflicts *prior* to packet transmission. The "Prior" solutions can be further classified into one of two categories.

    1. Static Prevention. All stations' permissible transmissions during every time-slot are dictated by a static transmission rights schedule, one copy of which is kept per station. The schedule is designed so that during every time-slot, at most one station would be permitted to send to a given station.
    2. Dynamic Avoidance. A station with a packet for transmission firstly broadcasts a transmission request for that packet. All stations maintain a log of transmission requests received from other stations. During every time slot all stations execute the same conflict-free scheduling algorithm. This algorithm takes the transmission-requests-log, and decides which of the packets may be transmitted (by their source stations). The algorithm is 'conflict-free' in the sense that at most one station would be allowed to transmit to a specific destination during every time-slot.

- Resolve conflicts *after* they occur, using one of the the "Detect and Retransmit if lost" strategies.

- Replicate receivers. The destination conflict problem could be eliminated if all stations were equipped with one data receiver, one receive buffer, and one receiver-to-receive-buffer-memory-module bus per WDM channel.

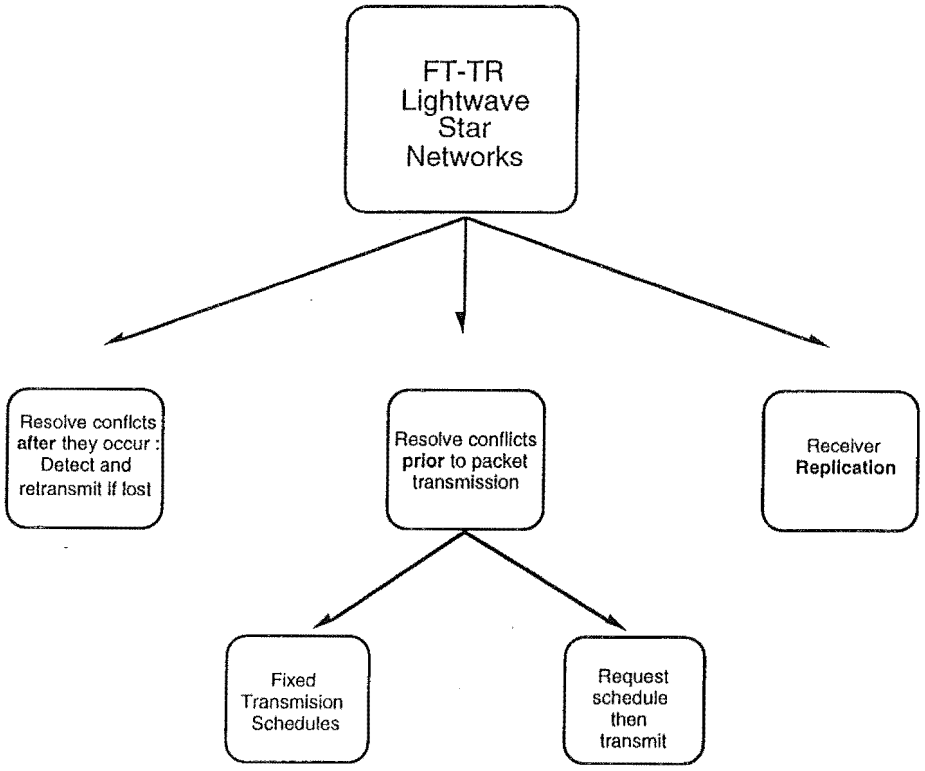


Figure 2.4: Classifying FT-TR networks according to their adopted strategy for destination conflict resolution

2.3 Resolving Destination Conflicts *Prior* to Packet Transmission

2.3.1 Fixed Assignment of Transmission Rights

In Fixed Assignment networks, the stations which may transmit and the destination that each station may transmit to (if it was allowed to transmit during that time-slot) are specified by a fixed transmission rights schedule. Destination conflicts are prevented by ensuring that the schedule allows at most one station to transmit to each destination during a time-slot. Fixed transmission rights assignment is therefore the extension of time division multiplexing over a multi-channel environment [ROUS93], [CHLA90], [CHLA88], [CHIP93].

Each station is equipped with a fixed transmitter (FT) and a tuneable receiver (FR). An N station network requires N data channels. Station S_i, i=1, ..., N, can transmit data packets on channel λ_i. Each station uses its FT for transmitting on its data channel, and its TR for receiving packets from any one of the other N-1 data channels.

Time is divided into fixed size slots whose duration equal the time required to transmit one (fixed size) packet. Time slots are grouped into frames of M slots. During each time-slot within a frame, a number of transmission permissions, are granted for packet transmission between source-destination pair (S_i,S_j). Up to N transmission rights may be allocated during one time-slot. This static assignment of transmission rights can therefore be defined by an $N \times M$ allocation matrix.

A destination-conflict-free allocation matrix for an N station FT-TR network presented in [CHIP93] is displayed in Fig. 2.5 (note that it was assumed that a station may transmit to itself). Each column represents the transmission permissions during a specific time slot in a frame of N slots duration. An element with the value of d in the (i,j)th position implies that station S_i would be permitted to transmit a packet to station S_d during the jth slot of every frame. A station therefore always knows which data channel to receive from during every slot by inspecting its copy of the allocation matrix. No pre-transmission signalling between the transmitting and receiving station is needed. We observe that the allocation matrix is destination conflict free, since the value of all elements in each column are distinct.

The problem of developing destination-conflict-free transmission matrices for FT-TR systems that maximises the system throughput under a given traffic pattern, was studied in [ROUS93]. Define $p_{i,j}$ as the probability that a packet generated at station S_i is destined to station S_j. The work of [ROUS90] addressed the problem of deriving an optimal schedule, given $p_{i,j}$, $i,j = 1, ..., N$ and the packet generation rates of every station. Also the authors proposed heuristics for finding near-optimal schedules.

	t	t+1	t+2	...	t+N
S_1	1	2	3		N
S_2	N	1	2		N-1
.					
.					
.					
S_N	2	3	4		1

Figure 2.5: A Destination-Conflict-Free Fixed Transmission Rights Assignment Schedule for Uniform Traffic

The fixed assignment of transmission rights have special advantages. Des-

tination conflict free operations are achieved, yet stations do not need to exchange control information. Inter-station co-operation necessary for avoiding conflicts is prescribed in advance, and encoded into the transmission schedule, one copy of which is kept by each station. The benefit is that no extra transceiving nor electronic processing is needed for exchanging control information. A station always knows which data channel to listen to during every slot. The sender need not inform the destination station in advance of packet transmission, nor the channel that the destination station should receive from. Another advantage over some solutions is that packet transmissions always succeed. As a result, stations need not bother with success monitoring nor acknowledgements nor retransmissions.

Notwithstanding, the fixed-assignment approach may not be suitable for some applications. Queuing delays experienced by packets can be unnecessarily high under light load or fluctuating traffic conditions. Let M be the length of a schedule frame. Analytic and simulation results of [BOGI93] showed that the average time that a packet has to wait (buffered) before transmission was at least $M/2$ time-slots long, although the traffic load was very low. With fixed-assignment protocols, a station must wait for the appropriate time-slot within the frame before it may transmit, even if no other stations in the system has a packet to be transmitted. The delay performance of fixed-assignment protocols deteriorates as the system size increases. In FT-TR systems, the minimum frame length for destination conflict free schedules is M=N-1 slots, N being the number of stations.

Fixed assignment protocols also handle bursty traffic inefficiently, especially under low load. A burst of H packets from a station addressed to the same destination would require at least $H(N-1)$ time-slots to transmit, even if all other stations were idle.

The drawback of a long wait before transmission may be alleviated by routing packets through intermediate nodes [CHLA90]. Suppose a ready packet at station S_i is destined for S_j, and according to the fixed schedule, it has to wait 20 slots before it is allowed to send to that destination. If the schedule allowed some station, say S_c to transmit to S_j just 10 slots from now, and that S_i is allowed to send a packet to S_c now, then station S_i's packet may be delivered sooner if it was first transmitted to S_c, where it would be retransmitted to S_j. Analogously, it may be quicker to reach a destination by taking several connecting flights through intermediate stops, instead of waiting for a direct flight, if direct flights are infrequent. This variation of fixed assignment introduces the issues of routing, and congestion control at intermediate nodes. If the chosen intermediate node is congested, then the packet may experience

an even longer delay. There is also a trade-off due to added propagation delay, and bandwidth use.

It has been noted that fixed-assignment networks are suitable only for highly regular traffic patterns [CHEN91], [CHLA91], [KAZO93].

2.3.2 Request-schedule-then-transmit

Another method proposed for resolving destination-conflicts *prior* to packet transmission, is through the "request-schedule-then-transmit" procedure, refer to Fig.1.2b.

"Request-schedule-then-transmit" networks [CHEN91], [CHIP93], [CHIP92], [CHEN92] prevents destination conflicts by requiring every station to establish a global view of all packets waiting for transmission in all stations [CHEN91], [CHEN92]. This view can be represented by an $N \times N$ backlog matrix B. Element $b_{i,j}$ indicates the number of packets at station S_i that are destined for station S_j. During every time-slot, all stations use their B matrix and the same scheduling algorithm to compute the same destination-conflict-free transmission schedule for the next time-slot. It was assumed that all stations are at the same distance from the star coupler, so the backlog matrices maintained by all stations are identical. Additionally, all stations execute an identical scheduling algorithm. Consequently, a station can deduce from the transmission schedule it computes during the current slot, which station (if any) is allowed to transmit a packet to itself during the next time slot. Thus the station would know which channel to receive from $2a+1$ time slots from now, using the transmission schedule that it had computed during the current slot.

The "request-schedule-then-transmit" protocols proposed in [CHEN91] and [CHEN92], implements this concept as follows. A network with N stations uses N data channels and one control channel. Stations are synchronised, and channels are time-slotted. The duration of each slot equals the transmission time of one packet, plus the time required by a receiver to tune from one channel to another, see Fig. 2.6. One data channel is devoted for data transmission of each station. The control channel is shared by all stations using TDM. Slots on the control channel are divided into N mini-slots, and one mini-slot is assigned to each station (Fig. 2.7). Let the station-to-star coupler delay equal a time slots.

Every station is equipped with one FT and FR for data exchange, and one FT and FR for transmitting and receiving from the control channel.

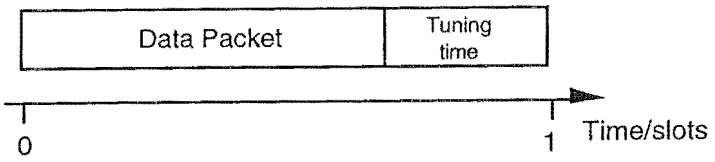


Figure 2.6: Data Slot of [CHEN91], [CHEN92]

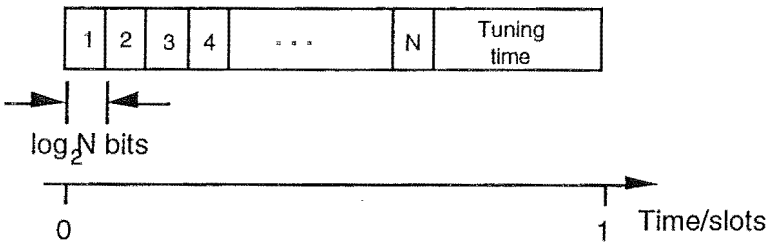


Figure 2.7: Control Slot of [CHEN91], [CHEN92]

The transmit buffer of each station is organised into N-1 queues of waiting packets, one queue per possible destination.

Packet exchange involves four steps.

Request: During every time-slot, if a new packet is generated at a station S_i, the station transmits the packet's destination address on mini-slot i (the mini-slot assigned to station S_i) of the current control slot.

The transmission of the packet's destination address on the mini-slot, constitutes a request to all stations to give S_i permission to transmit that packet.

Schedule: During every time-slot, each station receives requests contained in all mini-slots from the control channel, and use their information to update the station's copy of B. Then every station use B and the Maximum Remaining Sum (MRS) algorithm to compute a transmission schedule. Thus all stations will independently compute the same transmission schedule. A transmission schedule specifies which packets waiting in stations (whose requests have been received and noted in B) may be transmitted during the next time-slot. The MRS algorithm employs a heuristic for finding a schedule that allows as many packets as possible to be transmitted during one time slot, subject to the constraints that the transmitted packets have to have distinct destinations (so that trasmissions during that slot are conflict-free), and that a station can transmit at most one packet during one time-slot (since each station has one data transmitter).

Transmission: During every time-slot, a station transmits a packet destined for

a specific station, as specified by the transmission schedule it computed during the previous time slot, if that schedule allowed a packet to be transmitted from the station.

In general, more than one packet may be buffered at the station, waiting for transmission to the specific destination. Packets are logically organised into N-1 FIFO queues, where a new packet would be enqueued in the ith queue if it was destined for S_i. The packet that is transmitted would be the one at the head of the queue for that destination.

Reception: By examining the transmission schedule it computed $2a$ time-slots ago, a station can determine which incoming channel would contain a packet destined for itself during the current time slot. Since that schedule satisfied the conflict-free constraint, at most one packet would arrive for it during a time-slot. The station tunes its receiver to that channel and receives that packet (if any). Packet transmission and reception can proceed in parallel.

MRS was one of two conflict-free scheduling algorithms studied in [CHEN91]. Benchmarking of MRS using randomly generated B matrices suggested that MRS can find a near optimal transmission schedule, maximising the number of station packet transmissions subject to the destination conflict free constraint, in a relatively small number of operations ($O(N^2)$). The other algorithm reported in [CHEN91] is the System of Distinct Representative (SDR) algorithm. SDR can find an optimal schedule (i.e. one maximising the number of stations permitted to transmit a waiting packet), subject to the destination conflict free constraint. But SDR has higher computational complexity ($O(N^4)$).

Simulation analysis show that [CHEN91]'s conflict-free protocol could achieve near optimal utilisation. Furthermore, because the protocol prevents destination conflicts, once a station transmits a packet, its successful reception is guaranteed - no success detection nor retransmission is required. However, a packet must be delayed for at least $2a+1$ time-slots before it could be a candidate for transmission. During this period, a packet would be refrained from transmission even if all other stations are idle. Also, requiring all stations to maintain matrix B, as well as computing the transmission schedule during every time-slot, places high electronic processing demands on all stations.

A new packet must be stored in the buffer of its origin station during

1. the period when its destination address is being broadcasted to other stations. Thus each packet must be buffered for at least $2a+1$ time-slots.

2. Then the packet must be buffered during subsequent time-slots until it is at the head of its transmission queue and a schedule permits the station

to transmit it.

To reduce buffering costs, a buffer-sharing conflict-free protocol was proposed in [CHEN92]. The protocol follows an almost identical approach to solving the destination-conflict problem as the protocol in [CHEN91]. In addition the protocol aims to reduce buffer capacity required by stations, by lowering the buffer size needed for storing packets during stage 2, i.e. for storing packets whose presence have already being announced to other stations, and are waiting for a schedule that permits their transmission. It does so through a new common conflict-free scheduling algorithm which firstly assigns a maximum number of stations for conflict-free transmissions, and then assign the relocation of packets from congested stations to uncongested relaying stations for distributed buffer sharing.

In summary, the merit of "request-schedule-then-transmit" networks is that near optimal (100%) throughput can be achieved by resolving destination conflicts *prior* to packet transmission. Destination conflict free operation also mean that once a packet has been transmitted, its successful reception is guaranteed - no success detection nor retransmission is required.

"Request-schedule-then-transmit" networks also have their own weaknesses:

1. All new packets are forced to wait for at least $2a+1$ time-slots, even if all other stations were idle during that period.

2. Packets must be stored in a buffer during their announcement broadcast ($2a$ time slots), and then for the variable number of time slots until they reach the front of their transmission queue, and then wait for one or more time slots until a schedule permits their transmission. Buffer memory requirements may therefore be high. One solution proposed is to implement a packet distribution algorithm for routing packets from congested stations to uncongested ones [CHEN92]. The tradeoff in this case is an increase in the complexity of the electronic processing of each station.

3. Whilst they enjoy very high throughput, they are still sub-optimal in terms of throughput since

   (a) each station's backlog matrix B is at least $2a+1$ time-slots out of date; and

   (b) the MRS and RS scheduling algorithms are sub-optimal, whilst the SDR algorithm has very high complexity.

4. Complex buffer organisation. Each station either need to maintain N-1 FIFO queues in its transmit buffer [CHEN91], or be equipped with N-1 transmit buffers [CHIP93], one per possible destination.

In high-speed networks, the propagation delay time may be very compared with the packet transmission time. To illustrate, consider a MAN with a 30 km diameter. At the transmission speed of 1 Gbps, the propagation delay is 150μs. Assuming 512 byte data packets, the transmission time of a packet is approximately 10μs. This implies a propagation delay period 15 times that of the packet transmission time. Thus point 1) implies missed transmission opportunities of up to 16 time slots, and point 2) implies that every station must be provided with a buffer large enough to store at least 16 packets.

2.3.3 Hybrid Solutions

A variation of the "request-schedule-then-transmit" method was proposed in [CHIP92], [CHIP93]. Their protocol combines fixed conflict-free assignment with "request-schedule-then-transmit", to resolve destination conflicts *prior* to packet transmission. Hence it is called the Hybrid TDM (HTDM) protocol. HTDM alleviates the electronic processing burden of stations by interleaving fixed assignment (which does not require stations to compute schedules) with the "request-schedule-then-transmit" mode of operation.

Stations in HTDM networks are equipped with one FT and TR for data exchange, and one cFT and cFR for exchanging control information. N data channels and one control channel are required for an N station network.

Let the N data channels and the common control channel be time slotted, as before. Group $N + M$ data slots into a frame, where N is the number of stations (as before) and M is an integer such that N is divisible by M. The N slots of each frame are pre-assigned according to a fixed transmission assignment table [ROUS93], [GANZ94], [CHIP93], [CHLA87], [CHLA88], see section 2.3.1 on page 33. Destination conflicts during the N slots of each frame are prevented by designing the fixed assignment table so that it satisfies the constraint that every destination node is assigned to receive from at most one channel at a time.

Unlike the pure fixed assignment scheme, after every n slots, where $n = N/M$, one slot is left "open" [CHIP93] in which a station may transmit a packet to any receiver. However, stations' transmissions during this "open" slot must be destination conflict free. HTDM assures this by using a "request-schedule-then-transmit" method very similar to that proposed in [CHEN91]

[CHEN92], see section 2.3.2.

At each station, newly arriving packets destined for different stations are stored in separate first-in-first-out (FIFO) buffers. To implement HTDM, every station must have the queue state information of each FIFO-queue in all other stations.

At the beginning of an "open" slot, each station broadcasts the status of each of its buffers, taking into account the packets that will be transmitted in the pre-assigned slots during the next $2a$ time slots (recall $2a$ is the source-destination propagation delay when all stations are a slots from the star coupler). The status of each buffer can be empty or non-empty, so each station transmits N bits on its mini-slot of the control channel during a frame.

All stations listen to the control channel and use the received information to maintain their copy of the buffer status record of the buffers in all stations.

At the beginning of every open slot, every station executes a common algorithm, to determine which (if any) packet it should transmit during the slot. Thus all stations will independently obtain the same transmission schedule.

The common algorithm is called the Random Scheduling (RS) algorithm. The RS algorithm [CHIP93] uses an heuristic which selects packets randomly, checking each one with selections done in previous iterations, and accepting the packet for transmission only if there is no destination conflict.

Like [CHEN91], [CHEN92], a station can determine the channel that it should receive from during a time-slot, by an examination of the transmission schedule it computed $2a+1$ time-slots ago.

The primary advantage of HTDM is that stations need to compute a conflict-free schedule only once every n time-slots, instead of once every time slot [CHIP93]. However, its ability to adapt to varying traffic patterns is inferior to the pure "request-schedule-then-transmit" protocols of [CHEN91], and [CHEN92]. Also packets may be unnecessarily delayed from transmission by both the static assignment and dynamic scheduling components of the HTDM protocol. Static assignment means that a packet at the head of its transmission queue must wait until the first time slot when the schedule permits the station to send to the packet's destination, even if all other stations were idle in the meantime. Under dynamic scheduling, a new packet must be announced to all other stations, before it can be considered for transmission. Thus, a packet must wait between $2a+1$ and $2a+1+(N/M)$ time slots before it could be considered for transmission during an "open" slot. Each station in a HTDM network is fitted with N-1 FIFO transmit buffers. This may be a drawback when N is large.

2.4 Detect-and-Retransmit-if-Lost Networks

In networks using the "detect-and-retransmit-if-lost" technique, e.g. [CHEN90], [CHLA91] , [PAPA92], and [CHLA94], stations may transmit ready packets without first ascertaining the intentions of other stations. If two or more packets arrive for the same station during a time slot, all but one would be lost. Destination conflicts are detected and resolved only *after* they occurred.

2.4.1 Detect-and-Retransmit Networks Using a Common Control Channel for Signalling

Stations in the "Detect-and-Retransmit-if-lost" based network proposed in [CHEN90] are equipped with a FT and TR for data exchange, and a fixed tuned transmitter (cFT) and receiver (cFR) for control signalling.

An N station network requires N data channels and one control channel. Channels are time slotted, where the duration of each slot equals the transmission time of one fixed size packet plus the tuning period of the tuneable receiver. Slots on the control channel are divided into N mini-slots, see Fig. 2.8. One mini-slot is assigned to each station.

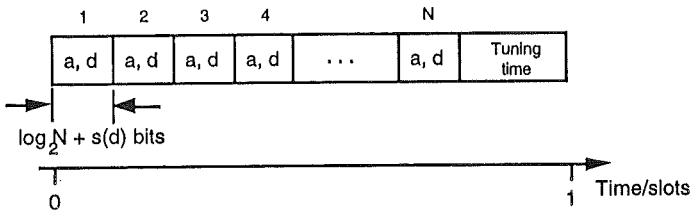


Figure 2.8: Structure of a control slot assumed in [CHEN90]

The exchange of data packets in this "Detect-and-Retransmit" network generally follows the procedure outlined in Fig.1.2a. Specifically, during every time-slot, a station executes the following steps.

Transmission A station, say S_i, transmits a new packet by writing the destination address of the packet, and its accumulated delay in the station's mini-slot of the current control slot. The packet is unconditionally transmitted on λ_i during the next time-slot.

Packet Reception Each station continuously monitors every mini-slot at the control wavelength. After examining all mini-slots received during time-slot t, station S_j knows if any data packets arriving during the next time-slot are intended for it. When more than one packet is destined for S_j, S_j selects one

of them for reception based on a commonly agreed arbitration procedure.

Arbitration The arbitration procedure proposed in [CHEN90] is: among the packets destined for itself, choose the packet with the largest accumulated delay for reception. All other packets intended for that station will therefore be lost.

Detecting unsuccessful packet transmissions Each station determines the outcome of the packets that it had transmitted by continuously monitoring the incoming control channel, and examining all mini-slots in every control slot. If its mini-slot in the control slot is not-empty, i.e. if the station has sent a packet one round trip to the hub delay period ($2a$ time slots) ago, then the station compares the destination address in its own mini-slot, with the (destination) addresses in all other mini-slots in the control slot. If none of the other mini-slots contain the same address, then the packet it sent to that destination would be successfully received. The sending station can find and purge the copy of that packet from its buffer.

On the other hand, if one or more of the other mini-slots contain the same address, a destination conflict occurred.

Retransmission If a station detects a destination conflict involving a packet it transmitted, the station must deduce whether its packet was received by executing the arbitration procedure described above. If the packet was lost, the station must find the copy of the packet stored in its buffer, and retransmit it as soon as possible.

This process has been depicted in the event-distance-time diagram of Fig.1.2a.

Figure 2.9 demonstrates the transmission procedure in a network with N=4 stations. All stations are assumed to be a=5 slots from the star coupler. In Fig. 2.9(a), station S_2 wants to send a packet to S_4. During time-slot t it transmits the address of S_2 , i.e. 2, and the accumulated delay of the packet (say, 7 time-slots), on its mini-slot in the current control slot. During the next time-slot, S_2 transmits the packet on its data channel, i.e. on λ_2.

Figure 2.9(b) shows the situation as the control and data packet of S_2 arrive at the intended destination, i.e. at S_4. S_2's mini-slot arrives first, during $t + 2a$. Notice that other mini-slots in that control slot may also contain information transmitted by their owners. S_4 receives all control slots. After S_4 decodes all addresses in the mini-slots of the $t + 2a$-th control slot, it can deduce that two packets intended for itself would arrive during the next time slot, because it had found its own address (i.e. 4) in the second, and

third (i.e. S_2 and S_3's) mini-slots of the current control slot. S_4 then invokes the arbitration procedure, which finds that the packet on λ_2 has the largest accumulated delay. Next, S_4 tunes its receiver to λ_2 during the tuning period of $t + 2a$ to receive that packet during $t + 2a + 1$.

Figure 2.9(c) shows the situation at S_2 $2a+1$ time slots after it transmitted the packet. The signals arriving at S_2 are identical to those arriving at S_4. To detect whether the packet that it transmitted $2a+1$ time-slots ago succeeded, S_2 receives all mini-slots. By examining all addresses in non-empty mini-slots, it finds that one other station also sent a packet to S_4. It therefore invokes the Arbitration procedure. Upon determining that its packet was chosen for reception by S_4, it then locates the copy of that packet kept in its buffer, and deletes it.

The signals arriving at S_3 during $t+2a+1$ are identical to those arriving to the other stations. To detect whether the packet that it transmitted $2a+1$ time-slots ago succeeded, S_3 receives all mini-slots. By examining all addresses in non-empty mini-slots, it finds that one other station also sent a packet to S_4. It therefore invokes Arbitration to deduce which packet had been chosen for reception. Upon determining that its packet was not chosen for reception by S_4 (i.e. lost due to a destination conflict), it then locates the copy of that packet kept in its buffer, and retransmits it following procedure Transmission.

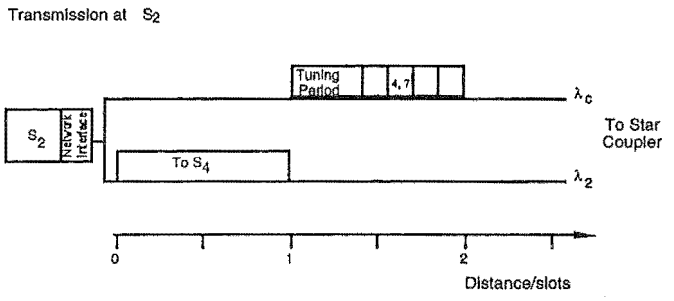
Detect-and-retransmit solutions permit stations to transmit packets with little delay, and without regard to the transmission intentions of other stations during the same time-slot.

This transmission procedure results in a very low *minimum* excess packet delay. Numerical results show that mean packet delay is near minimum under light traffic. However, destination conflicts may occur, requiring success detection by transmitting stations, and packet retransmission. Thus the average packet delay can be very high at high loads. For a ten station network where all stations are a=5 time-slots from the star coupler, mean packet delay exceeds 45 time-slots, when the offered load is 0.6 packets per station per slot. Also the need for retransmissions limits throughput to at most 63%. Packets must be buffered at their source station for at least $2a+1$ time slots. Stations therefore require a transmit buffer capacity of at least $2a+1$, which grows with network diameter or transmission speed.

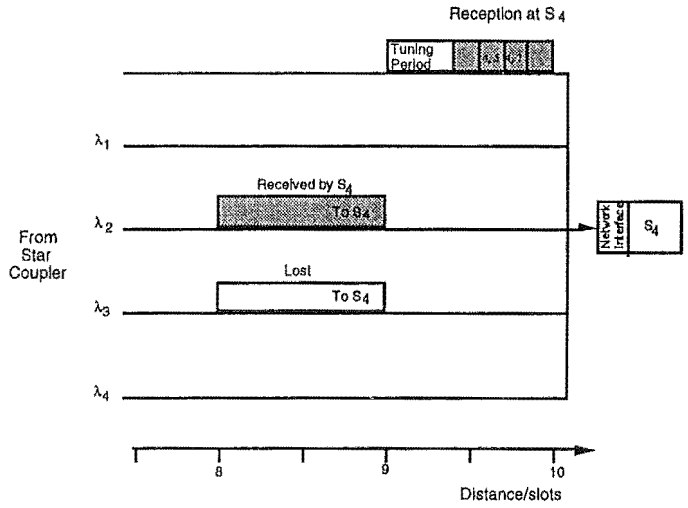
Several other networks have being proposed which is based on this "detect-and-retransmit-if-lost" principle. In the network proposed in [PAPA92], in addition to the DT-WDMA MAC procedure, each station executes a Receiver Collisions Avoidance Learning Algorithm (RCALA) which decides about which of the packets waiting for transmission at the station will be transmitted at the beginning of the next time slot. Another variant of the "detect-and-retransmit" network uses multiple control channels for pre-announcing packet transmissions [HUMB93]. If one control channel is dedicated to each station, then it can be shown that during a time-slot, each station only needs to process control information relating to packets destined for itself, providing that stations that want to transmit to it have already established a virtual connection [HUMB93]. Finally, the probability of packet loss can be reduced by providing each station with multiple opportunities to receive incoming packets by equipping each station with multiple tunable filters for switching packets into delay lines for later reception [CHLA91], [CHLA94]. These enhanced "detect-and-retransmit" networks are considered next.

2.4.2 Detect-and-retransmit Networks Where Stations use a Receiver Collision Avoidance Algorithm Using Multifeedback Learning Automata

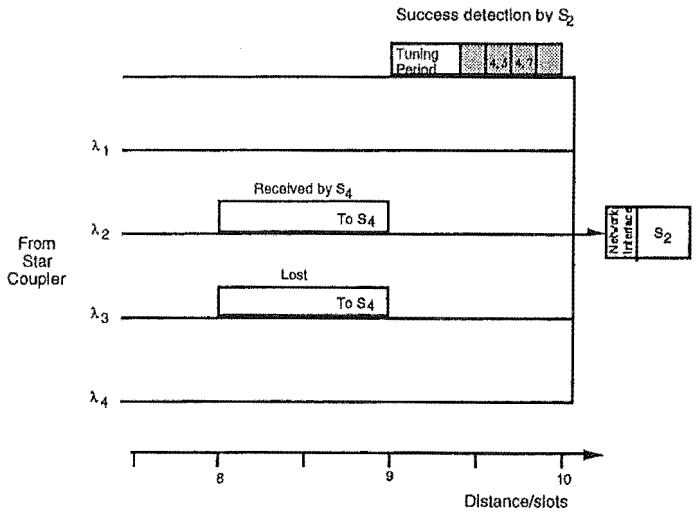
In the DT-WDMA (Detect-and-retransmit) network considered above, a station decides which packet it should transmit during the next time slot following the FIFO discipline. That is, it chooses the first-to-arrive packet in its buffer which either has not being transmitted, or which has being transmitted but



a) Mini-slot and packet transmission by S_2 during time-slot t for sending a packet to S_4.



b) Mini-slots and packet reception (indicated by shading) by $S4$ during t +2a +1 time-slots later



c) Success/failure detection by $S2$ t +2a+1 time-slots later

Figure 2.9: Main steps for the successful transmission of a packet from S2 to S4 in the "detect-and-retransmit" network of [CHEN90]

was detected to have being lost. Let us refer to packets which either have not being transmitted, or which have being transmitted but were detected to have being lost as *ready* packets.

If a station has ready packets destined for more than one station, then one expects that the probability of a successful packet transmission can be improved if the station smartly chooses to send a packet which is destined for a station which is likely to be free to receive the packet. Thus, instead of transmitting ready packets in FIFO order, the station chooses the packet which is more likely to be received.

In [PAPA92] an improved detect-and-retransmit network was proposed, where each station is provided with a learning automation which decides about which of the packets waiting for transmission will be transmitted at the beginning of the next time slot. The learning automaton used is a multifeedback automaton, specially designed for the destination conflict avoidance problem of FT-TR networks. Each station maintains N-1 FIFO queues of ready packets, one queue per destination. Each station S_i maintains a probability set P(t) = $\{P_{i,1}(t), P_{i,2}(t),, ..., P_{i,r}(t)\}$ where $P_{i,j}(t)$ is the probability weight corresponding to destination S_j during time slot t+1, and r is the number of destinations of the ready packets at S_i during t.

Let $D_i(t) = \{S_k \mid$ the k-th queue is not empty$\}$. Then the probability of station S_i selecting each destination is defined as

$$\begin{aligned} r_{i,j}(t) &= P[S_i \textit{ transmits to } S_j \textit{ at time slot } t] \\ &= \frac{P_{i,j}(t)}{\sum_{k \in D_i(t)} P_{i,k}(t)} \end{aligned} \tag{2.1}$$

A station implements the selection rule by dividing the (0,1) interval into k subintervals, with the width of each subinterval corresponding to the probability of selecting each destination station. The station then use a uniform random number generator to produce a random value in (0,1). The value is therefore located in one of the subintervals. The destination selected by the station is the one which corresponds to the subinterval which contains the value. Having selected the destination, the station can identify the packet which should be transmitted during the next time slot as the ready packet which is the first-to-arrive amongst the packets which are destined for the selected destination.

A receiver collisions avoidance learning algorithm (RCALA) is executed by all stations during each time slot to update the $P(t)$ probability set. RCALA is

based on the use of Multifeedback Learning Automata, see Fig. 2.10. Loosely speaking, RCALA examines the outcomes of the transmissions of all stations during time slot t, by analysing the contents of the control slot received during $t+2a$ (the transmission announcements by stations for packets that are transmitted during t reaches all stations during $t+2a$). RCALA then adjusts the transmission probability set assuming that stations where a destination conflict did not occur are more likely to be free. Thus weights corresponding to destination-conflict-free destinations are "rewarded" by having their probability weight increased, and weights corresponding to destinations where a conflict had occurred are "penalised" by being reduced.

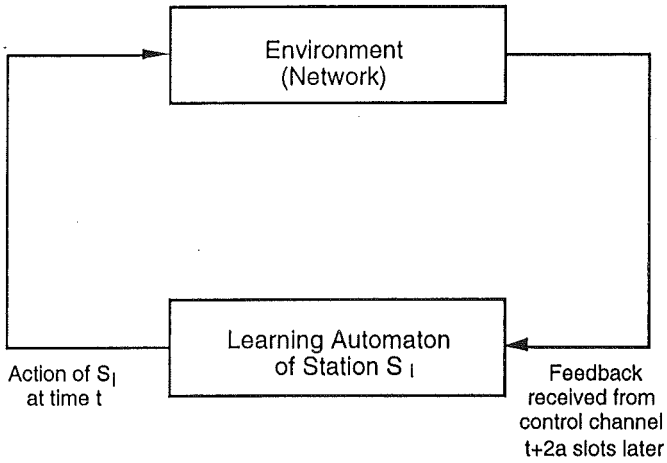


Figure 2.10: An Automaton whose actions trigger environmental responses, and then use a learning algorithm to take into account the responses when it decides which actions it would take next, to improve the probability of choosing actions that best match its goal.

Results of [PAPA92] confirmed that the network performance can be improved somewhat, when RCALA was used. However, the *extent* of improvement was limited. In the cases considered, RCALA networks achieved a throughput of approximately 63%, versus 59% for DT-WDMA. A similar degree of delay improvement was observed.

It should be noted that a (the propagation-delay-to-hub to transmission time ratio) was assumed to be 2 or 4 in the networks studied. Since the feedback on the outcomes of transmissions during t is not available to the RCALA algorithm used by stations until $t+2a$, it is unclear whether an increase in a would degrade the performance of these multifeedback learning automata based networks.

2.4.3 Detect-and-Retransmit Using Multiple Control Channels

In [HUMB93], a variation of the "Detect-and-Retransmit" method is proposed for reducing the amount of control information that needs to be processed by a station during a time slot.

Each station is provided with a tuneable transmitter (cTT) and a fixed tuned receiver (cFR) for control purposes, in addition to a FT and TR used for data packet exchange. Each FT and cFR in the network is assigned a unique channel. Therefore, for a network with N stations, $2N$ channels are needed. N of them are for data transmissions and are called data channels. One data channel is dedicated for the data transmitted from each station. The remaining N are for control purposes, and are called control channels. One control channel is assigned to each station. The cFR of a station is fixed tuned for receiving from the control channel assigned to it.

The control channel of a station, say S_j, is used by source stations to signal to S_j the fact that they intend to transmit a packet to S_j. The bandwidth of the control channel of S_j normally must be reserved by stations that wishes to send packets to S_j. The proposed protocol distinguishes between three classes of network applications, each of which has a corresponding procedure for accessing the control channels. Class 1 refers to connection-oriented traffic with dedicated (fixed) bandwidth allocation. Class 2 is connection-oriented with dynamic bandwidth allocation, such as virtual-circuits. Class 3 is datagram traffic, and was expected to be rarely used. Bandwidth is allocated dynamically for Class 2 and Class 3 traffic, and the problems of destination conflicts on the data channels, and collisions on the control channels may occur. The majority of network applications were expected to generate class 2 traffic [HUMB93]. We therefore describe the solution to these problems in the context of Class 2 traffic (connection-oriented with dynamic bandwidth allocation).

Time is divided into frames of length T. Each control frame is divided into $m+d$ slots. m slots of each control frame may be used for control signalling for connection oriented traffic. The remaining d control slots of each control frame are dedicated exclusively to signalling for datagram traffic. The data channel frame is divided into $n+1$ slots, where n slots are used to transmit the data packets and one slot is used as a *status slot*. Each station uses its status slot for transmitting the assignment of slots for its control receiver.

The packet transmission procedure consists of 2 phases : the Connection Setup phase, and the Data Transfer phase.

Setup Phase

Suppose station S_i wants to setup a connection with S_j. The following steps are executed.

1. S_i tunes its TR to the wavelength of the data transmission channel of S_j, listens to the status slot, and locates idle (unassigned) slots in frames of S_j's control channel.

2. S_i picks one of these idle slots in the control frame of S_j and transmits a connection request on the control channel of S_j using its control transmitter. At this time, S_i also informs S_j of all the idle slots in the frame of S_j's control channel that S_i can use.

3. On receiving the connection request from S_i, S_j assigns a slot in its control frame that S_i can use. S_j then updates the information sent in its status slot to indicate the assignment of a slot on its control channel to S_i. S_i learns of this assignment by tuning to S_j's data channel and monitoring the status slot there.

4. S_j tunes its data receiver to the data channel of S_i, listens to the status slot, and locates the idle slots in the control channel of S_i.

5. S_j picks one of the idle slots in the control channel of S_i, and transmits a connection acknowledgement on S_i's control channel using its (tuneable) control transmitter. S_j also informs S_i of other idle slots in the control frame of S_i's control channel that S_j can use.

6. On receiving the connection acknowledgement from S_j, S_i assigns the slot picked by S_j in its control frame to S_j. S_j learns of this by tuning its data receiver to the data channel of S_i and monitoring the status slot there.

There may be a collision in step 2), if another station S_k picked the same idle slot in the frame on S_j's control channel that S_i picked (say slot 18). Also, this slot need not be idle since S_i's knowledge of the idle slots in S_j's receiver is $2a$ time-slots old, and in the meantime S_k could have been assigned slot 18. If S_k uses it at the same time as S_i, there will be a collision. The collision is detected if S_i's address does not appear in the status slot of S_j's data channel after a round-trip delay. S_i recovers from collisions by randomly picking an idle slot on the retry.

Data Transfer Phase

Suppose station S_i wishes to send a packet to station S_j, and that a full-duplex connection has been established between the stations. S_i picks a slot at random from the n slots in its next data frame, say slot number 4, and announces this slot in one of the idle slots in S_j's current control frame using its cTT (i.e. transmits '4' on one of the idle slots in S_j's current control frame). During the next frame S_i would send the packet on its data channel in slot number 4. After one source-to-destination propagation delay period, S_j would receive all control slots on its control channel, including the one used by S_i. After decoding that control slot, and assuming that no other stations transmitted control information on the same slot, S_j would know that a packet would be destined for itself from station S_i on the 4th slot of the next data frame.

This transmission succeeds if

1. no other station used the same idle slot in the control channel of S_j, and
2. no other station chose to transmit to S_j on the same data slot (slot number 4) on their data channel.

If 1) is violated then a *collision* has occurred on the control channel of S_j, and both packets would be lost. If 2) is violated, then a *destination conflict* occurred, and all but one of the packets involved would be lost.

When a destination conflict occurs, the destination involved picks one of the packets for reception using some fair mechanism, and the other packets destined for that destination would be lost. Destination stations are tasked with acknowledging the receipt of (successful) packets. Acknowledgements are piggy-backed onto data packets. Hence sending stations can detect the loss of its packet by the absence of an acknowledgement. However acknowledgements could also could be lost, and so provision is made for this by the use of timers.

For packet switching applications, the advantage of this multi-control channel "detect-and-retransmit-if-lost" method is that each station need only process control information relevant to itself.

The obvious drawback of the multi-control channel approach is that an extra tuneable device is needed per station. Thus the network interface of each station requires both a tuneable transmitter and a tuneable receiver. It has been noted that one intended benefit of allocating a unique data channel to every station is that both a tuneable transmitter and a tuneable receiver would not be required [BOGI93].

Collisions on the control channels, and destination conflicts on the data channels may lead to high packet delay and reduce throughput. Moreover, N channels are dedicated for control use. Considering average packet delay, a station have to wait for one to n slots after signalling, before packet transmission.

The efficiency with which this protocol handles demand-assigned bandwidth/connecti oriented traffic (i.e. class 2) improves with n, as defined by

$$P_s = 1 - (1 - \rho/n)^m \tag{2.2}$$

where P_s is the data slot utilisation, and ρ is the offered load. For this reason n needs to be large for efficiency, but this means a larger average packet delay for connectionless traffic. The maximum throughput for connectionless traffic is limited by that of slotted ALOHA.

For Class 2 traffic, the throughput and mean packet delay can approach that of the DT-WDMA protocol of [CHEN90], assuming that a connection has already been setup. The advantage over "detect-and-retransmit-if-lost" networks using one control channel is that a station has to process control information of packets destined for itself and not for all other stations. This enables the electronic processing requirements of stations to be reduced.

2.4.4 Detect-and-retransmit with Multiple Reception Opportunities

An interesting improvement on the Detect-and-retransmit scheme is to provide every station with multiple opportunities to receive incoming packets [CHLA91], [CHLA94], thereby significantly reducing the probability of packet loss.

In a network named QUADRO (Queuing Arrivals for Delayed Reception), each station is equipped with d+1 fast wavelength tuneable optical 2×2 switches, and d delay lines at their input port [CHLA94]. These are used by the station for receiving one of the packets involved in a destination conflict, and optically queuing the others sequentially. Fig. 2.11 details the delay line and switch layout for the input port of a station.

In QUADRO networks, N data channels and one control channel is required for an N station network. N data channels are assigned to stations, one each. Like [CHEN90] and [PAPA92] an additional channel, called the control channel, is shared by all stations using TDM. Channels are time slotted,

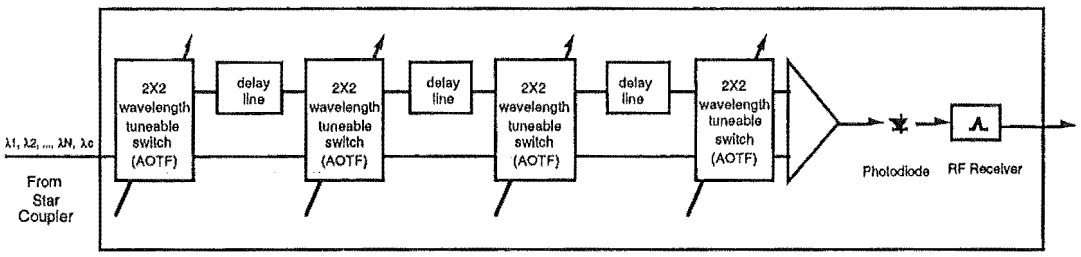


Figure 2.11: Switched Delay Line used by stations in QUADRO networks

the control channel being further divided into N mini-slots.

Every station is equipped with a cFT and cFR for sending and receiving from the control channel. Every station also has one FT and TR for data exchange.

A station may transmit a packet by writing the destination address of the packet in the mini-slot reserved for that station during the current time-slot, and then transmit the packet on its own data channel during the next slot. Hence the minimum packet delay prior to the transmission of a packet at the head of the transmission queue is just one time-slot.

If two or more packets from different stations arrive during the same time-slot for the same destination, say station S_j, a destination conflict occurs at S_j. When S_j is equipped with d+1 switches and d delay lines, there is a reception window of d time slots immediately after the packets' arrival to S_j. The station may switch one of the packets during any time-slot within the packets' reception window to its data receiver for reception, provided the station does not have to receive another packet during the same time slot. Packet loss occurs only when there are more than d+1 packets involved in a destination conflict, or when too many destination conflicts occur during consecutive time slots. The decision algorithm used by a station for deciding which packets in its delay lines to switch for reception was called the reception strategy.

All sending stations monitor the control channel and execute the same reception algorithm as the destination station. As a result, a source station could determine if the packet(s) it transmitted has been successfully received. The origin station must retransmit lost packets.

Simulation results [CHLA91] showed that throughput can be increased from 62% without QUADRO, up to 85% or 95% if QUADRO is used where each station was fitted with d=10 delay lines and 11 wavelength tuneable

optical 2×2 switches. The extent of throughput gain varies between 85% or 95% depending on the way stations select packets for transmission, and on the policy used by stations for receiving packets. Transmission strategies include the random selection of a packet from the station's input queue for transmission, and FIFO transmission. A reception strategy specifies the procedure for deciding which packet in a station's delay lines to switch to its receiver for reception. Reception strategies considered include LIFO, FIFO, and Oldest Packet in the Receiver (OPR). The later of course require the receiver to know the cumulative delay experienced by the packet, thus the age of packets should also be transmitted by the sending station on its mini-slot.

QUADRO allows stations to transmit new packets without waiting a long request broadcast and schedule computation period [CHEN92], and enjoys low packet loss probability (5% to 15% with 10 delay lines per station). Performance benefits of QUADRO therefore are a high throughput (85% to 95% compared to 62% for DT-WDMA), low packet delay, and a near optimal minimum packet delay. However, the feasibility of this solution depends on the costs of equipping all station with multiple wavelength sensitive switches and delay lines.

Each of the d 2×2 wavelength selective optical switches requires a rapidly tuneable filter. In [CHLA94] the acoustically-tuneable optical filter (ATOF) was suggested as the wavelength selective component of the 2×2 switch. But the ATOF is the dominant cost component of a tuneable receiver. The need for d of such switches per station may limit the economic feasibility of the QUADRO solution.

If the filters have a non-ideal characteristic, then the selection of one packet for reception may weaken the signal power of the remaining packets in the delay line. The power loss due to this effect accumulates as a packet traverses the d delay lines.

Another opportunity for improvement is that in QUADRO packet loss must still be resolved by the sending station. Hence success/failure detection and the associated electronic processing overhead, and packet retransmission is necessary.

2.5 Replicate Receivers

A frontal attack on the destination conflict problem is to provide each station with N optical receivers for data reception [GOOD89], [KOBR87], [SENI91].

In the LAMBDANET prototype [GOOD89], [KOBR87], each station has one fixed tuned transmitter (FT) for transmitting packets. For data reception, each station has a wavelength demultiplexer (e.g. a diffraction grating) for separating the incoming optical signal into their individual wavelengths, followed by N optical receivers.

An N station network uses N channels, one for data transmission by each station. Each station may receive and process all its information asynchronously and in parallel, with all other stations on the network.

Packets may be exchanged between stations as follows. Each station transmits data on a unique wavelength using its FT. Each station must receive all packets (on any of the incoming channels) and decode their headers to select those destined for itself.

This solution to destination conflicts has the merits of optimum throughput and near minimum packet delay. The obvious drawback is that each station requires multiple fixed receivers. The number of receivers required in a network grows as N^2, which may be prohibitively expensive, and in general would provide a receiving capacity far greater than needed. In addition each station requires at least N-1 receiver buffers and N-1 busses for transferring packets from these buffers. If the packet transfer bandwidth between the MAC layer and the upper (LLC) layer is W packets per slot, then the MAC layer requires extra buffer space to hold received packets, until they could be forwarded the station's LLC layer, whenever $N > W$. There may also be delay and queuing at the destination due to the processing there.

Advances in optoelectronic integrated receiver arrays, however, may mitigate these disadvantages.

2.6 Conclusions

Through the parallel use of multiple channels by stations, WDM multiplies the fraction of fibre's huge bandwidth that can be employed.

This chapter introduced a classification of WDM star networks based on whether the data transmitters and receivers of their stations are fixed tuned or rapidly tunable. Each class of networks has its specific MAC problems. Fig. 2.12 illustrates the classification, and the main problems faced by each class of networks.

In this chapter, WDM network architectures and protocols that appeared in the literature were reviewed, with emphasis on FT-TR networks. The *main*

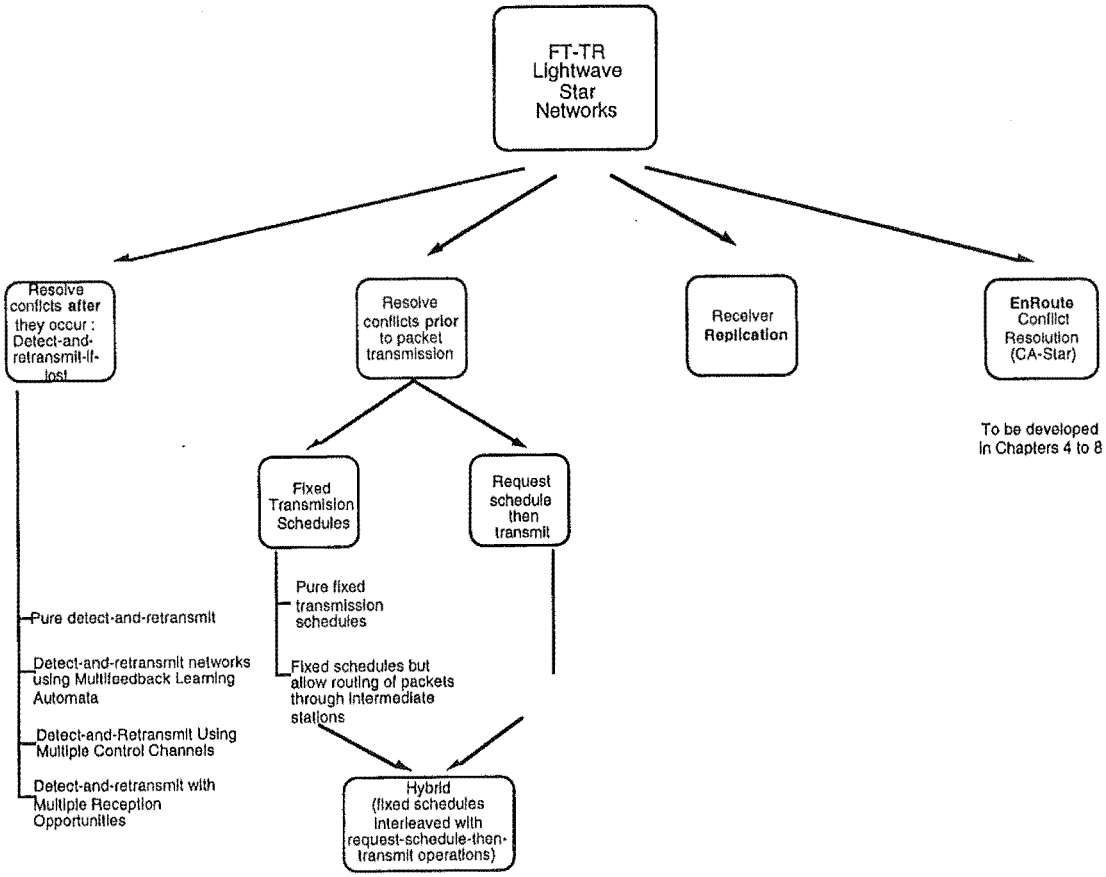


Figure 2.12: Classifying FT-TR networks according to their strategy for destination conflict resolution.

challenge in FT-TR networks is the problem of destination conflicts, which must be solved before the potential of the networks could be realised. When multiple data channels are simultaneously used, more than one packet may arrive for a station during a time-slot, causing a destination conflict. If that station has just one receiver, it can receive only one of the incoming packets. The other packets would be lost.

It was shown that in "fixed-transmission-schedule" and "request-schedule-then-transmit" networks, the problem of destination conflicts is resolved *prior* to packet transmission. Alternatively, in "detect-and-retransmit" networks, stations are charged with detecting and resolving conflicts *after* they occurred. Finally, in "receiver replication" networks, the destination conflict problem can be solved by providing each station with multiple data receivers. The classification of FT-TR networks according to their adopted strategy for

destination conflict resolution has been summarised in Fig. 2.12.

Chapter 3

WDM Star Network Technologies and Assumptions

As mentioned, the CA-STAR networks considered in this thesis belong to the FT-TR (fixed tuned transmitter - tuneable receiver) class of WDM star networks. This chapter firstly describes the broadcast-and-select star network architectural form. Secondly, the main devices assumed for the construction of FT-TR networks, and the assumptions on their characteristics, are reviewed. Current technologies that are relevant to FT-TR networks are then discussed.

3.1 Broadcast-and-select Star Lightwave Networks

A WDM broadcast-and-select star network comprises N stations, and one star coupler. Every station is connected to the passive star coupler by two fibres as shown in Fig. 3.1. One fibre of each station carries signals transmitted by the station to an input port of the star coupler. This is called the station's *outgoing* fiber. At the star coupler, inputs from all stations are combined and broadcasted to all outputs. The second fibre of each station is used for carrying signals from an output port of the star coupler to the station. This is called the station's *incoming* fiber. For logical clarity, we will represent station i by two separate blocks: the transmitting module and the receiving module, see Fig. 3.1.

A station accesses its outgoing and incoming fibres through its network interface.

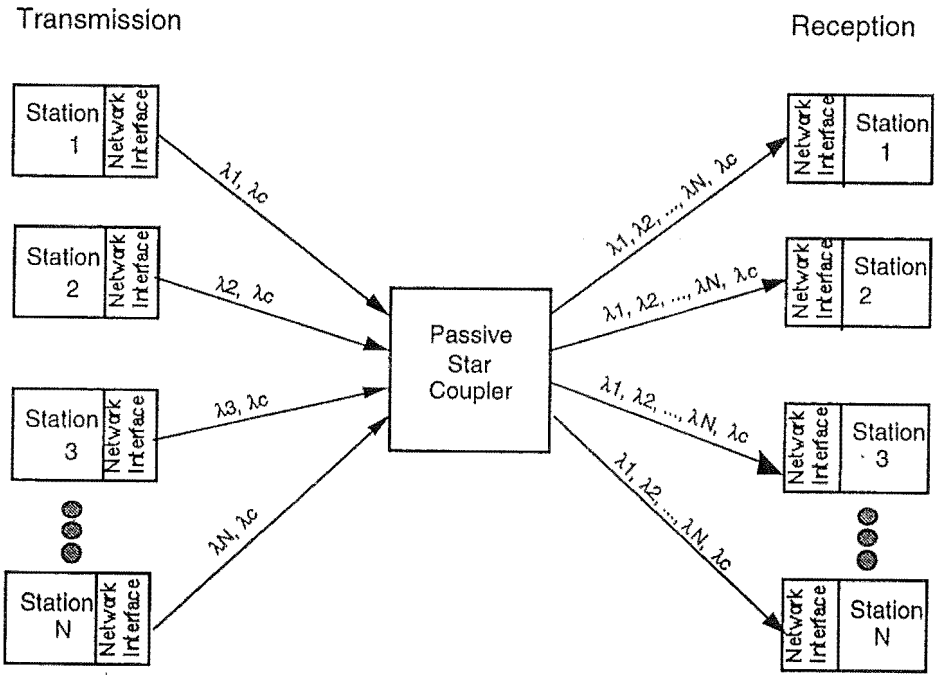


Figure 3.1: WDM Star network with N stations

The main technologies assumed by FT-TR networks relate to the availability of

(T1) single frequency lasers (FTs),

(T2) fixed tuned filters (used in fixed tuned receivers, FRs) and fast tunable filters (used in tunable receivers, TRs), and

(T3) the star coupler.

The above devices are assumed to have the characteristics which permit the construction of WDM multi-access networks. To access the low loss regions of fiber, the single frequency laser sources are assumed to be able to be centered at a given wavelength in commercially used fibre's low loss windows near the 1.3μm and 1.55μm regions. Fixed tuned filters are also assumed to be available. It is assumed that they could be centered at a given frequency within one of the 1.3μm or 1.55μm windows. The tuneable filters are assumed to have a reasonably fast tuning speed, say form several tens of ns to a few μs. They can be tuned between any pair of center wavelengths within the low loss windows in this time. When commercially available, their cost however, is expected to be considerably greater than the fixed tuned devices. Another

assumption is that the characteristics of lasers and filters allow a minimum channel spacing equal to a few times the bandwidth of the channel.

Alternatively, in the case of local area networks, the 0.8μm window may be of interest in view of the existing experience and investment in technologies suited for operation in this region. Although the attenuation of signals within this window in fiber is higher, it may not be a critical drawback for the shorter distances expected for local area networks.

A motivation for WDM networks is that the bit rate that can be achieved by each station is limited, and the achievable rate is far less that which can be supported by optical fiber. One reason is that the upper limit for lasers is in the region of 10Gbps. They are also power limited, the maximum being typically 5 mW. It is feasible to use at most a few laser sources and receivers per network interface, typically smaller than the number of channels in the system. Likewise it is preferred that the number of agile optical components (e.g. tuneable filters or tuneable sources) per station be small, because of their higher cost relative to fixed tuned devices. These assumptions imply that the number of channels which can be employed for packet switching would be in the thousands, given rates of 1-10 Gbps per channel. Furthermore, the fast tuning speed of tuneable filters implies that a tuneable receiver can receive consecutive packets from different channels. A corollary of these assumptions is that more channels would be available than the number of users. Hence the WDM systems being considered will not be bandwidth limited. Instead the rate determinant is the network interface which can support a rate of at most a few tens of Gbps. Finally, the ability to use time slotted channels was assumed in many FT-TR networks, e.g., [PAPA92], [CHEN90], [CHEN91], [CHEN92], [CHLA91], [MEHR90], [CHIP93], [HUMB93]. Thus network stations are assumed to have synchronised clocks.

It has been generally assumed that the cost of agile devices would be high compared with fixed tuned ones, so that the number of tuneable devices per interface would be a major cost determinant. Hence in most WDM networks proposed (see Chapter 2), each station is equipped with only one tuneable filter for channel selection, in addition to fixed tuned laser sources and fixed tuned receivers.

3.2 Technological Considerations

The choice of the star topology, and the assumptions on device characteristics appeals very much in view of the limitations and the current progress in

photonic technology.

3.2.1 Choice of Topology

With current devices, there are two aspects related to the electronic bottleneck faced by lightwave networks. The first of these, the limited power budget of laser sources, is better accommodated by the chosen star topology than traditional linear bus, folded bus, and ring topologies.

To detect a transmitted bit, a receiver requires a minimum amount of optical energy, E. Even in the ideal (shot-noise-limited) case the level of power arriving at a receiver must be at least EB, where B is the bit rate. For instance, consider the most power conservative network topology, namely a two station network connected by a point-to-point fiber link. For typical semiconductor laser devices, their power level P is approximately 1 mW, and E is approximately 500 photons per bit. In this case, even assuming a lossless connection, the power constrained throughput is approximately 10 Tbps, well less than the bandwidth limited throughput of about 50Tbps. That is, the network capacity is constrained by a power bottleneck, and not bandwidth limited.

In real networks with an arbitrary number of stations, this power bottleneck becomes more severe. With N stations we have a choice on the topology for their interconnection, the limited power budget being a major influence on the selection. Bus and ring topologies have been actively studied by researchers and adopted in most of today's LAN and MAN implementations. Their advantages include low cable layout, and an imposed ordering of nodes which permit efficient, collision-free and destination-conflict free access protocols to be developed [DYKE88], [FRAT81], [JOHN87], [KIM90], [BANE92], [CONT91], [DRAV91], [PACH95], [WEN94]. Accordingly most protocols in use or proposed such as token ring, Ethernet, DQDB, Express-Net, and FDDI/FDDI II are based on ring or bus structures.

Nevertheless the power bottleneck restricts the progression of these topologies into high-speed photonic implementations. Bus and ring structures suffer from the power related drawbacks of splitting loss and excess loss, both worsening as the number of network stations increases. Splitting loss results from sharing (broadcasting) the signal power of a packet amongst stations, so that only a fraction of the transmitted power is actually available to the intended destination. The devices used for collecting and splitting the signal power amongst stations are not ideal. The losses of power due to the non-ideal characteristic of the devices in the network fabric is referred to as excess loss.

Splitting Loss

Consider the bus topology configured for unidirectional transmission on one segment and reception on the other, e.g. in D-Net [TSEN83]. Users interface the transmit side of the bus by means of directional optical couplers, and receive from the reception side using a second set of couplers. This first problem is that with N stations only

$$\alpha^2(1-\alpha)^{2N-2} \tag{3.1}$$

fraction of useful power is delivered in the case of the farthest transmitter-receiver pair in the network, where α is the fraction of the transmitted power coupled onto the bus (i.e. the line-out coupling coefficient of each tap) when there is no excess loss. With N stations, the splitting loss (in dB) is proportional to 2(N-2). Differentiating w.r.t. α, setting to zero, solving for α, and substituting, gives the optimum fraction of power received

$$\frac{1}{(2N-1)^2}\left(1-\frac{1}{2N-1}\right)^{2N-2} \tag{3.2}$$

For large N, this becomes

$$\frac{1}{4eN^2}. \tag{3.3}$$

In reality the power-limited throughput would be lower than that suggested previously due to excess loss associated with the fibre couplers or taps used to collect and distribute optical signals. In bus and ring topologies the excess loss (in dB) is proportional to N [HENR89], [RAMA93]. For example, the excess loss between the two farthest stations is

$$-10(2N-1)log_{10}\beta \tag{3.4}$$

in the folded bus case, where β is the fraction of the total input power to the coupler (tap) that appears as the total output power.

Power limitations led many researchers to investigate alternative topologies, a common one being the passive broadcast-and-select star [ACAM89], [BRAC90]. As before let us assume that signal power P is available from each of the N transmitters. The star coupler uniformly distributes the combined inputs across its N output fibres, so each output delivers to its receiver a share of P/N from each transmitter. Thus the splitting loss (in dB) is proportional to $log_{10}(N)$. A given transmitter-receiver pair can exchange data at a power constrained rate of $P/(NE)$. The maximum aggregate throughput of the N station network would therefore be at most P/E. This is the same as the two

station case we considered earlier on, but with $N/2$ times the transceiving potential. Still it compares very favourably over the ring and bus topologies which had a maximum throughput of P/EN, as seen from Eqn. 3.3.

Excess Loss

The advantage of star over bus and ring topologies is further increased when we account for the excess loss associated with physically realisable optical couplers and splitters. Combining and dividing of optical signals occur only at the passive star coupler in a star network. The most mature technique for combining N sources (each at a unique wavelength) and dividing the resulting composite signal among N outputs (to receivers) is by multistage interconnection of the evanescent-field 3-dB 2×2 directional coupler shown in Fig. 3.2.

A coupler simply consists of 2 optical waveguides. Each waveguide is brought close together over a coupling distance of length L, so that a fraction of the power from the input of each waveguide can be transferred to the other[1]. For building a star coupler the power splitting ratio (coupling coefficient) is usually 0.5, so that almost half of the light at the input of either waveguide will appear at each of their outputs.

These couplers are not perfect and exhibits excess loss. As a result the power level of each of the two output ports would be under half of the level incident to each of the two inputs.

In general, the power splitting ratio can be set between 0 and 1. This property, combined with their simplicity, have led to other uses for directional couplers in previous WDM star networks. The power splitting ratio can be set either by varying the relative diameter of each waveguide (i.e. changing d_1 and d_1, Fig. 3.2), or by changing the coupling distance (L), or by changing the refractive index of the material [CHIN95], [HWAN95]. When used in packets switched networks, directional couplers are fixed to a specific power splitting ratio setting. The reason is that the mechanisms for changing these parameters limit the speed at which the device could be reconfigured for a new splitting ratio. The speed of electronics required to control them also limits the speed at which the device could be reconfigured from exhibiting one splitting ratio to another.

A $N \times N$ star coupler can be built by interconnecting the 2×2 couplers. Generally we can construct a transmissive $N \times N$ star, with N equalling a power of 2, using $N/2log_2(N)$ 2×2 couplers interconnected so that each

[1]Traditionally, this is done by fusing the core of the two fibers over the coupling distance

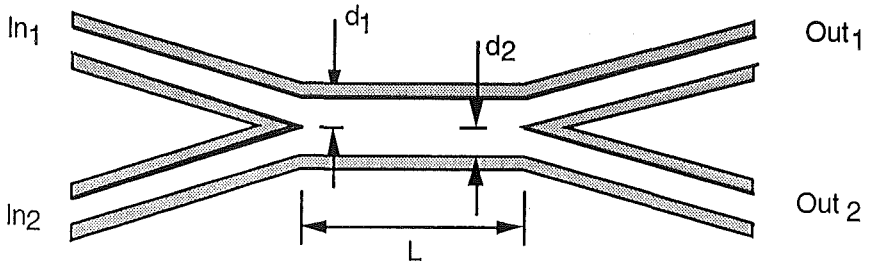


Figure 3.2: 2×2 Directional Coupler

incident input traverses $log_2(N)$ such 2 ×2 couplers before arriving at one of the output ports. The excess loss suffered by a signal traversing the structure is proportional to the number of stages, and hence the number of elementary couplers. Thus, the excess loss would be proportional to $log_2(N)$. This is another win of the star over the bus and ring topologies where excess loss increases linearly with N.

One disadvantage of the star topology compared with bus and ring topologies is the star's high expected cable requirements [LIN89]. However recently efforts have been directed at reducing the star's cable lengths. For example, a reflective $N \times N$ star constructed from a mix of ordinary 2×2 couplers, and from reflective 4×4 star structures have been proposed. This development enables a given star network to be implemented using half of the fiber required by the existing technique.

Another development provides an alternative means of realising the $N \times N$ star coupler, in a way that avoids the complexity for interconnecting a large number of basic couplers [DRAG89]. The proposed $N \times N$ coupler is constructed in free space using two arrays, each comprising of $N \times N$ elements. It was shown to have a theoretical efficiency that is independent of N, and efficiencies exceeding 35% are obtainable in practical applications [DRAG89].

3.2.2 Single-frequency Laser Sources

A number of laser devices presently available meet the demands of single-frequency operation of assumption T2) [LEE91a], [BRAC90], [LEE89], [GLAN91], [KAZO90], [CHEN95].

Conventional lasers based on the Fabry-Perot (FP) resonating cavity exhibit a multi-mode spectrum. The modal losses in FP lasers is independent of frequency, so many modes can reach lasing threshold.

To modify the FP laser for producing single frequency output, a frequency

dependent loss element, typically a corrugated structure, is built inside the laser cavity with a period selected so that all modes but one (corresponding to the desired frequency) is suppressed from oscillation. The wavelength which experiences constructive interference (least destructive interference) is called the Bragg wavelength. Devices producing single frequency output based on this concept are called Distributed Fraby Bragg (DFB) lasers [LEE91a]. Numerous variants of DFB lasers have been investigated in the past years, including Buried Hetrostructure (BH), Etched Mesa BH (EM-BH), Self-Aligned constricted Mesa (SACM), and semi-insulating Fe-doped IP based devices [CHAW92], [GLAN91], [LEE91a], [BRAC90].

In addition to single frequency operation, performance characteristics include the degree of frequency excursion [LEE91a]. Frequency excursion or chirp results from the direct current modulation of single-mode lasers through carrier injection, causing an expansion of the modulation bandwidth of the optical signals. Direct current injection is used in both the On-OFF Keying (OOK) and Frequency Shift Keying (FSK) coding formats that are applicable for WDM systems. FSK modulation can achieve smaller channel spacing. This is due to a more compact spectrum resulting from the use of a smaller modulation current. Nevertheless, OOK demodulation is usually simpler and less demanding on the filter. Frequency chirp causes crosstalk given an imperfect filter response. The combined crosstalk and chirp effects limits the effective channel utilisation of a direct modulation WDM system. Several simulation and analytical studies have been performed to gauge its impact [WILL90], [LITO92]. For example, with each channel operating at 2Gbps, it was shown by simulation [LITO92] that chirp effects would require a minimum channel spacing of 37 GHz if OOK was used, and a spacing of 10 GHz if using FSK.

3.2.3 Optical Filters

Assumption T2 relate to the tuning speed, and the number of channels supported by optical filters. These characteristics depend on their tuning range, minimum channel separation, crosstalk, gain and distortion characteristics, which in turn depend on the principle used to distinguish the wavelength of interest, and specific technology used to apply it [CHI95], [SMIT89], [KOBR89], [CHEU89], [CROS93], [LOPE93], [HINK93], [OGUS93].

Most optical filters create wavelength selectivity by means of some interference phenomena, applied to distinguish the wavelength of interest. They differ in the mechanisms used as a source of interference, and the speed in

which the source can be changed for selecting a different wavelength

On the fixed tuned filter front, FP filters [LIU95], [OGUS93] may be suitable when tuning is not required as in assumption 3). They have a narrow linewidth (approx. 0.01nm), and also a medium tuning range (approx.50nm). However tuning is performed by changing the length of its resonating cavity, or by adjusting the cavity's angle to the incident beam, both being done mechanically. Thus FP filters tune too slowly for packet switching applications. Its tuning speed is typically several ms, whereas the packet transmission time is expected to be about several μs.

Active semiconductor filters tune using refractive index change created by current injection. Hence they can be tuned quickly, in the order of ns. Their ability to tune rapidly between two incoming channels, each on a different centre frequency simulating the operation of packet switching in WDM broadcast-and-select star networks have already been demonstrated, where each channel operated at 1 Gbps. Nevertheless active semiconductor filters have a very narrow tuning range (approximately 1-4nm), with a typical bandwidth of 0.05 nm. Thus they would support at most a few tens of channels.

Similarly, electrooptic filters tune rapidly (several ns) but also have a narrow tuning range (approx. 10 nm) and a broader bandwidth (approx. 1nm) so they can tune over approximately 10 channels [LOPE93], [CROS93], although recent developments based on the use of a synthesised grating structure may allow a wider tuning range [NOLT95].

Acousto-optic tuneable filters (ATOFs) are based on the same principle as the electro-optic devices [CHI95], [HINK93], [QINH95], [CHEU89]. It consists of a narrowband polarisation mode converter, sandwiched between crossed polarizers, see Fig. 3.3. The incident beam is converted to one state of poloarization, say the TE (horizontal) mode by the input polarizer, when entering the polarization converter. Narrowband TE to TM conversion is achieved using an anisotropic material such as lithium niobate. Within such a birefringent medium, different refraction indices are seen by the TE and TM modes, so these modes become different in their phase velocities, with TE and TM modes falling in and out of phase over an interaction length that depends on the frequency (and the material birefringence). In order to achieve narrowband conversion, a periodic stress is applied, producing successive TE-to-TM conversion and reconversion over a length relating to the drive frequency. When the drive frequency is chosen so that successive TE-to-TM conversion of the selected frequency are in phase, there would be constructive interference, and the outgoing wave would have a large magnitude in TM mode, i.e. phase matching is achieved for the selected frequency. The TM component is then

chosen by passing the light through the output (vertical) polarizer.

In acousto-optic tuneable filters, the perturbations are generated by means of a travelling acoustic wave, whereas with electro-optic devices this is done using an electric field. Both have similar bandwidth, but the acousto-optic filter has a much broader tuning range[2], specifically, the entire 1.3 to 1.56 μm range, and has a tuning speed of typically several μs. Thus it could support many hundreds to a few thousands of 1Gbps channels, depending on source characteristics, and modulation technique.

It has been suggested that a tuneable filter with fast tuning speed and wide range can be simulated using two or more acousto-optic tuneable filters. The concept involves operating several filters in a relay manner, so that while one filter is being used to select arriving packets on a given wavelength, the other filters are in the tuning phase [CHEN90], [CHLA94]. This method appears to be applicable, provided that the wavelengths containing the next M packets are known in advance, where M-1 is the number of filters employed. Also issues of power loss, or of switching the input signal between filters may arise.

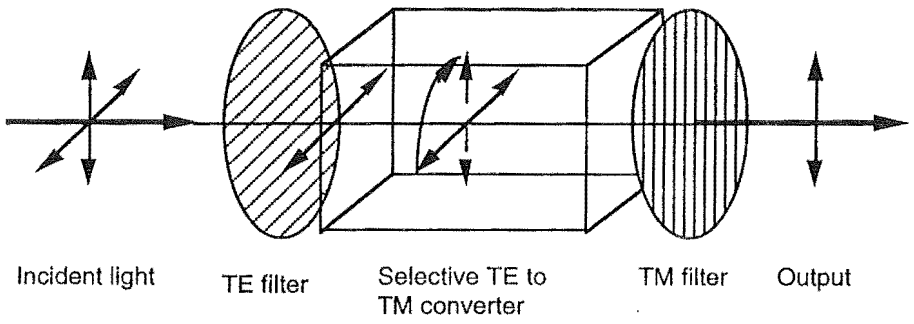


Figure 3.3: Mode coupling based tuneable filter

3.2.4 Photonic Amplifiers, Optical Switches, and Wavelength Converters

In addition to couplers, other devices that may be part of the network fabric include photonic amplifiers, optical switches, optical delay lines, and wavelength converters.

In WDM networks, photonic amplifiers and gain equalisers may be used to compensate for the power losses due to splitting loss and excess loss. The

[2]Another difference is that the AOTF can simultaneously select more than one wavelength

problem of the best location for the placement of optical amplifiers within a WDM star has been considered in [GREE94]. Additionally, photonic amplifiers (POAs) may also serve as ON-OFF optical switches.

Considering optical switches, the 2-state (ON-OFF) optical switch provide the simplest functionality: in the OFF state, the input incident light is absorbed; in the ON state, the incident light is passed [MAO95], [YAMA95]. They have been available for some time, and are also used in some network fabrics, photonic interconnects [OBRI95], and large arrays of ON-OFF switches are also a component of some tunable filters [SCHU95]. An 2-dimensional array of ON-OFF switches can function as a reconfigurable grating, and therefore is a means of achieving fast tunability.

N-way digital optical switches allow an incident optical signal to be directed to one of N outputs [CAVA91], [NELS94], [NELS94a] [JAGE94]. Refer to the PAC network (reviewed in section B.4 on page 242) for an example of their application in WDM star networks.

Lastly, some recent WDM networks assume the availability of photonic wavelength converters [BARR95], [KOVA95a], and [LEE95a].

3.2.5 Synchronisation

The ability to create slotted channels have been assumed in many previous WDM star architectures, e.g. [PAPA92], [CHEN90], [CHLA91], [HUMB93], [CHEN91], [CHEN92], [CHIP93], [CHEN94]. Synchronisation can be maintained by exchanging clocking data over the network, or by distributing a clock signal to all the stations [CHEN90], [CHEN91]. When adding a station to the network, the station uses its unique wavelength to measure its propagating delay to the star coupler [CHEN90]. Stations adjust their clocks relative to the star coupler clock.

3.3 Chapter Conclusions

This chapter introduced the main devices that are required for the construction of FT-TR WDM networks. Existing photonic technologies for realising some of the required devices were reviewed.

The main assumptions made by the WDM star networks in the FT-TR class relate to the availability and characteristics of T1) single frequency laser sources with narrow line-width, T2) fixed tuned filters, and most critically,

tuneable filters and their pass-band, fast (but non-zero) tuning speed, and a wide tuning range, and T3) a passive star coupler of a sufficient capacity (I/O ports) to support hundreds of stations without prohibitive excess loss.

Given the wide tuning range of the receivers and reasonably narrow channel spacing, a corollary is that more channels would be available than the number of users. Hence the WDM star network will not be bandwidth limited. Instead, the rate determinant is the speed of the electronics at its end stations which can support a maximum rate limited to a few Gbps.

Developments in photonic technology were reviewed, and the characteristics of the available devices relevant to the FT-TR class of networks were discussed. In particular, the operational principles of currently available devices which can provide the functionality demanded by (T1) and (T3) were reviewed.

The realisation of FT-TR networks still await the development of tunable filters that meet the requirements of (T2), i.e. filters which have both a wide tuning range and a fast tuning speed. It has been suggested that a tuneable filter with fast tuning speed and wide range can be simulated using two or more acousto-optic tuneable filters that are already commercial available. The concept involves operating several filters in a relay manner, so that while one filter is being used to select packets arriving on a given wavelength, the other filters are in their tuning phase. This method appears to be applicable, provided that each station always knows which wavelengths contains packets destined for itself during the next M time slots, where M-1 is the number of filters employed.

Chapter 4

sCA-STAR Networks

As mentioned in Chapter 1, three central arbiter designs are considered in this thesis (sCA, optCA, and rcCA, see Fig. 1.6), thereby originating three CA-STAR architectures. This chapter presents CA-STAR networks implemented using the sCA Central Arbiter, called sCA-STAR networks.

Section 4.1 introduces the sCA-STAR architecture, describing the network interface of the sCA and ordinary stations, and their interconnection. As discussed in section 1.4, two protocols will be proposed. The first, named sCA/B, has the bounded packet delay property, and maintains the ordering of packets transferred between any given source-destination pair, but packets may occasionally be lost due to buffer overflow at sCA. A connection acceptance control procedure must therefore be applied to keep the load to a level such that the probability of packet loss is acceptable. Section 4.2 is dedicated to the definition and performance analysis of the sCA/B protocol.

The second protocol considered for sCA-Star is called sCA/R. This protocol uses a procedure called reflection for guaranteeing delivery of packets. The description and performance study of sCA/R is contained in section 4.3. The final section focuses on the main findings and implications.

4.1 sCA-STAR Architecture

The logical architecture of a CA-STAR network based on a shared-memory Central Arbiter (sCA-STAR) is depicted in Fig. 4.1. It consists of N stations S_i, for i =1, 2, ... , N, and a shared-memory central arbiter station (sCA) located at the passive star coupler.

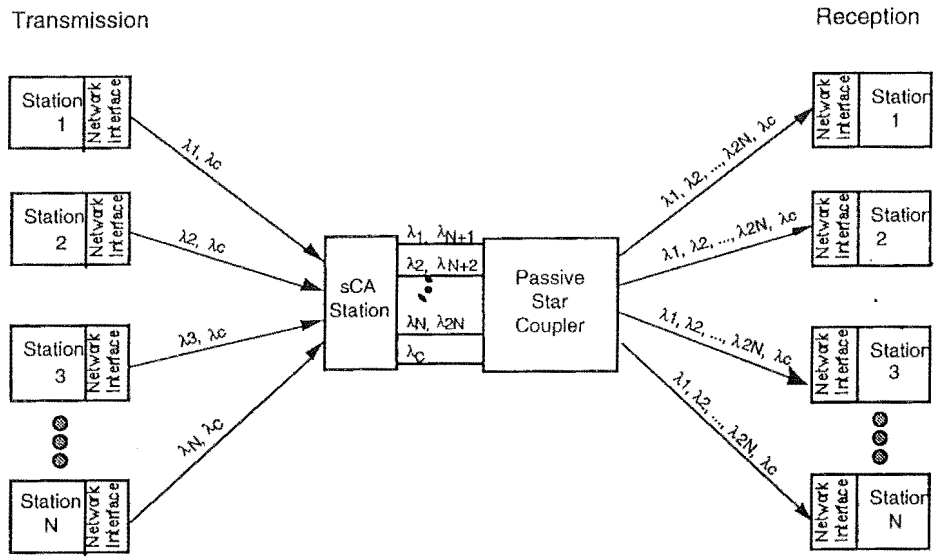


Figure 4.1: Logical architecture of an sCA-STAR network.

Each S_i is connected directly to input port I_i of the sCA by means of an optical fibre, called the *outgoing fiber* of S_i. This fibre carries signals from S_i to sCA. The sCA has N+1 outputs, one for carrying data from each station, plus one for sending control signals. These are connected to the input ports of an $(N+1)\times(N+1)$ star coupler. The star coupler combines optical signals from its N +1 input ports and broadcasts them to its N +1 output ports, as explained in section 3.2.1. Another fibre runs from each output port O_i of the star coupler to each S_i. It carries the combined signals from sCA to S_i.

4.1.1 Channel Structure

Stations and sCA are synchronised, and channels are time-slotted. The duration of a time-slot equals the transmission time of one (fixed length) packet, plus the tuning period [CHEN90], [CHEN91], [CHLA91], [CHEN92], [PAPA92], [HUMB93]. An sCA-STAR network with N stations would use $2N$ data channels, λ_1, λ_2, ..., λ_{2N}, and 1 control channel, λ_c. Slots on channels λ_1, λ_2, ..., λ_{2N} are called *data slots*. Each data slot can carry one data packet, see Fig. 4.2. Slots on the control channel are called *control slots*.

Each control slot is subdivided into N mini-slots plus a recall field, see Fig. 4.3. Mini-slot i (i=1,2, ..., N) can carry the address of one station. The recall field is N bits wide[1].

[1]The use of the data slots, control mini-slots, and the recall field, and the network interface of stations and sCA depends on the choice of MAC protocol, and is therefore

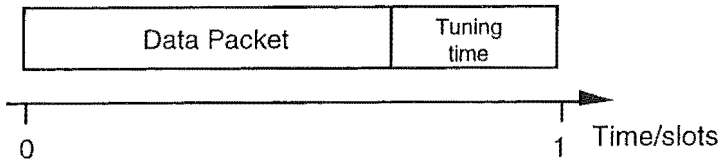


Figure 4.2: Structure of a data slot.

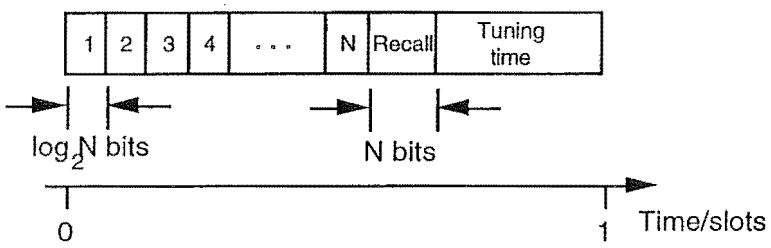


Figure 4.3: Structure of a control slot.

A station accesses channels on its incoming and outgoing fibres through a network interface.

4.1.2 Network Interface of User Stations

The network interface of station S_i operates two fixed tuned transmitters, see Fig. 4.4. One transmitter (FT_i) is for data transmission on channel λ_i. Station S_i uses channel λ_i (i =1, ..., N) as its own channel, dedicated for its data transmissions. The other transmitter (FT_c) is tuned to the common control channel, λ_c.

Prior to transmission, packets are stored in the transmit buffer of S_i. Ordinary stations need only a tiny transmit buffer, sized for storing up to three packets.

Station S_i has one fixed receiver for receiving from the control channel (FR_c), and one tuneable receiver filter for receiving data packets (TR). The TR can be tuned within the tuning-period of a time-slot (Fig. 4.2), for receiving from any one of the $2N$ data channels. The data and control receivers operate concurrently, therefore a station can simultaneously receive from the control channel and from one of the data channels.

---

explained in section 4.2

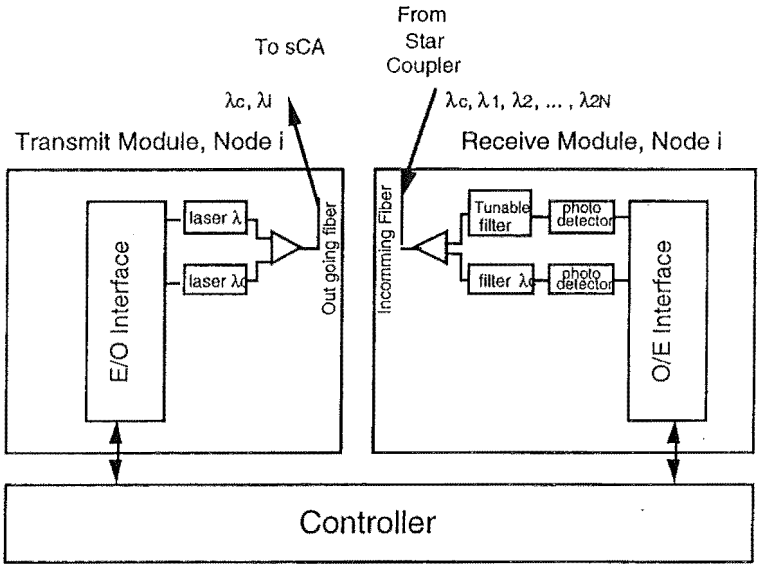


Figure 4.4: Block diagram of the network interface of ordinary stations in a sCA-STAR network.

4.1.3 The sCA Station

Construction

The sCA station is equipped with $N+1$ fixed tuned transmitters T_i and receivers R_i, $i=c$, 1, 2, ..., N, and a buffer. Fig. 4.5 illustrates the configuration of an sCA for a network with N stations.

The ith input to sCA (I_i) carries data on λ_i and control signals on λ_c. These are separated using a wavelength demultiplexer.

Signals on λ_i are split using a directional coupler, with $N/(N+1)$ fraction of the input signal's power directed to output port O_i, where it is merged with data transmitted on λ_{N+i} transmitted by T_i, see Fig. 4.6. The remaining $1/(N+1)$ of the signal power of λ_i enters R_i.

Likewise, signals on λ_c from all inputs are merged before being split, with $N/(N+1)$ fraction of the signal's power directed to output port O_{N+1}, where it is merged with control signals transmitted by T_c on λ_c, see Fig. 4.5. The remaining $1/(N+1)$ of the signal power enters R_c.

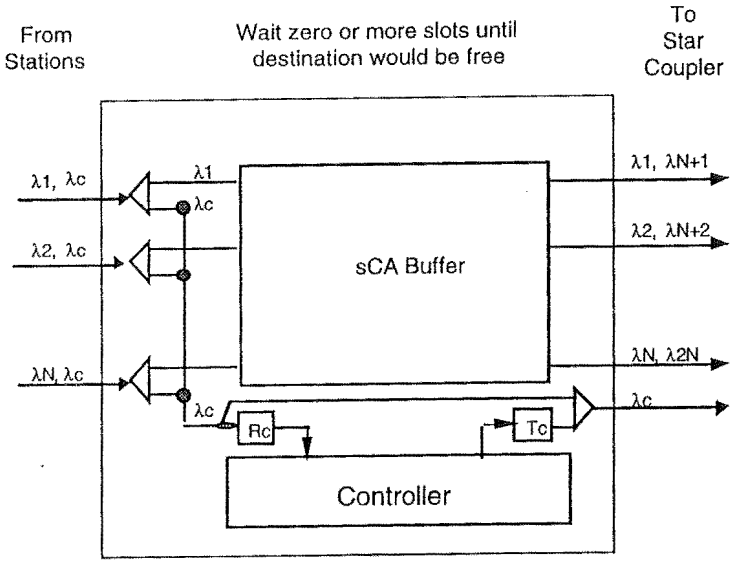


Figure 4.5: The sCA Conflict Arbiter station configuration for an sCA-STAR with N stations.

Logical Organisation of the Buffer Memory at sCA

From above, $N/(N+1)$ of the signal power of transmitted packets will always enter the star coupler. Those packets would be received by their intended destinations unless they are involved in a destination conflict.

sCA identifies packets transmitted by S_i which would otherwise be lost, and receives them using receiver R_i ($1/(N+1)$-th fraction of the signal power of packets transmitted from station S_i is directed to R_i). The rescued packets are stored in the buffer of sCA. Then sCA transmits the packets to their destinations when their destinations are free to receive them.

Let the memory space of sCA's memory be divided into packet sized units, each of which has a unique memory *address*. These packet sized memory spaces are logically partitioned into two fixed size regions. One region, called *tempbuff*, is used for storing newly received packets. Newly received packets are usually "transferred" from the tempbuff into the *central buffer*, where they wait for transmission to their destinations.

The memory space of the central buffer is shared by N (output) FIFO queues of packets. A packet destined for S_j is placed in queue j when joining the central buffer. Packets waiting in queue j are transmitted to S_j in FIFO order, using T_j.

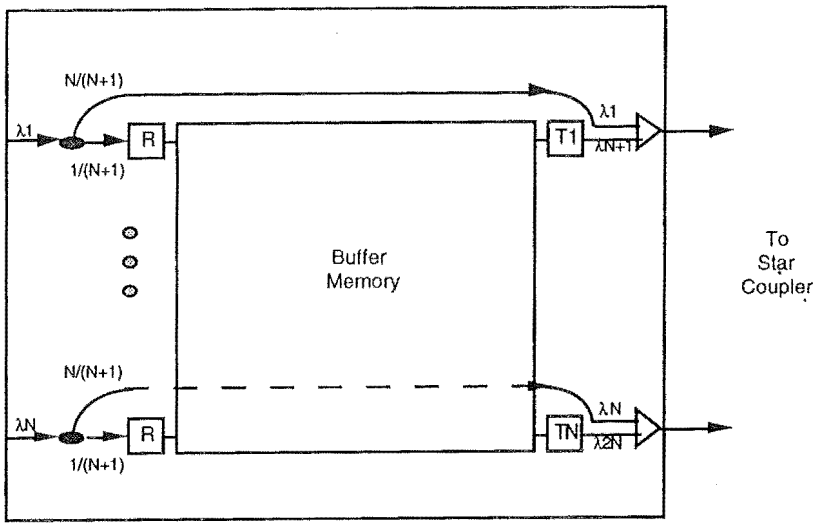


Figure 4.6: Paths of data signals in sCA.

Memory Management

Memory is dynamically allocated to each region, thereby avoiding the need for physical transfer of packets from tempbuff into the central buffer. The memory space of each region is defined by a list of memory addresses allocated to it.

A packet is "placed" into queue i of the central buffer by 1) adding its memory location to the tail of the ith queue, and 2) reimbursing tempbuff for the transferred space by adding one free memory address of central buffer to tempbuff's memory address list.

Queues and lists are logical structures defined by a chain of pointers to the addresses of their elements. If the memory space of sCA is represented as an array of packet sized locations, then think of a pointer as an integer representing the address of (pointing to) one location. A queue is defined by a sequence of pointers to elements in the queue. Adding to a queue means adding the address (an integer) of the new packet to the queue's pointer sequence. Use of pointers makes the physical transfer of packets unnecessary. Consequently, if a packet is received into one memory location of sCA, it remains in that location until read for transmission from sCA.

Likewise, removing a packet from a queue (e.g. after the packet was transmitted from sCA) simply involves deleting the pointer (one integer) from the head of the queue, and adding a pointer that points to the location that was occupied by the packet (i.e. whose value equals the address of the memory location occupied by that packet) to the free space list.

The memory space of central buffer is fully shared by all queues. We assume that the memory of sCA permits N concurrent read and write operations, providing they are performed on mutually exclusive addresses.

4.1.4 The Function of the Central Arbiter Station.

The central arbiter station (sCA) rescues packets that would otherwise be lost due to destination conflicts. A packet from S_i bypasses sCA if it would be successful. Otherwise, it is received by R_i, and waits in sCA until it can be transmitted to its destination without a conflict. All packets rescued need to wait for at least one time-slot. Hence the one slot delay for O/E conversion of a packet involved in a destination conflict is *desirable* delay. sCA transmits rescued packets to their destinations as soon as their destinations are free to receive them.

The relative efficiency of the various star network architectures depends on

$$a = \frac{station-to-star-coupler\ propagation\ delay}{packet\ transmission\ time}.$$

In optical MANs and LANs, high transmission rates and distances up to tens of kilometres are expected [GREE93]. In these environments $a > 1$, so one expects that significant performance improvements in terms of throughput and packet delay can be obtained with the CA-STAR architecture, as demonstrated in Fig. 1.4. This is due to the fact that in "detect-and-retransmit" networks, a packet experiences an additional propagation delay of $2a+1$ time-slots each time it is lost, plus the one source-destination propagation period. In CA-STAR even if a packet is involved in a destination conflict, its delay due to propagation still equals one source-destination period. Since CA transmits packets it had rescued in a conflict-free way, multiple retransmissions are not necessary. Thus, significant throughput improvements are also expected.

An advantage of CA-STAR over "request-schedule-then-transmit" networks is the resolution of destination conflicts without having to wait for a long request-and-schedule phase. In a CA-STAR network, new packets can be transmitted almost as soon as they are generated. Only packets that would otherwise be lost are "rescued" by sCA, and even so they are delayed at CA only until their destination is free to receive them (Fig. 1.4c). In "request-schedule-then-transmit" networks, a packet must wait until it is at the head of its transmission queue, and then wait at least $2a+1$ time-slots during its request broadcast, and then wait until a schedule permits its transmission (Fig. 1.4b). CA-STAR is thus expected to yield a reduction in mean packet

delay of at least $2a+1$ time-slots. Furthermore, when $a > 1$ CA-STAR is expected to have higher throughput than "request-schedule-then-transmit" networks, since the information used for scheduling in the later would be $2a+1$ slots outdated.

4.2 CA-STAR Protocols

4.2.1 Terminology

The use of multiple data channels implies that several data slots are available during one time-slot. Moreover, stations are not usually not equidistant from sCA, so not all stations 'see' the same data slots/control slot during a time-slot. Let us introduce some notation for referring to time/data/control slots.

Time

Let t denote the t-th time-slot (t=1, 2, 3, ...) where the duration of a time-slot equals the transmission time of one packet, plus a tuning period (see Fig. 4.2). The tuning period equals the time needed for tuning a tuneable receiver from one data channel to another [CHEN90], [PAPA92], [HUMB93], [CHEN91], [CHLA91], [CHIP93].

We assume that stations are time synchronised, so the beginning of each t is observed by all stations. t may be referred to as the "tth time-slot", or "slot t". Since the symbol t is reserved exclusively for denoting the tth time-slot, we shall also refer to the tth time-slot as simply t.

t denotes a specific *duration* of time equal to one time-slot. Sometimes it is necessary to refer to a point in time. Let us refer to the time instant at the beginning of time-slot t as "at the start of t". Likewise "at the end of t" refers to the instant at the end of the t-th time-slot.

Data and Control Slots

Given station S_i and time-slot t, define *incoming data slots* to be the set of slots on the $2N$ data channels arriving (on S_i's incoming fiber) to S_i during t. Define the *ith incoming data slot* to be the data slot arriving at S_i during t on λ_i, and the *incoming control slot* to be the control slot arriving at S_i during t. These definitions extend to the *outgoing* data/control slot(s) in the

expected way.

4.2.2 Structure of the sCA-STAR Protocols

The set of rules followed by network entities for transceiving on the transmission media constitutes its MAC sublayer protocol.

Ordinary stations of an sCA-Star network are freed from the burden of destination conflict resolution, so their MAC protocol could be simplified. Thus, in order to define a MAC protocol for sCA-Star networks, one needs to specify i) the protocol for ordinary stations, and ii) the protocol for sCA.

4.2.3 MAC Protocol for Ordinary Stations

The MAC protocol for ordinary stations consists of a Transmission procedure, a Reception procedure, and an Arbitration procedure. The Transmission procedure specifies the procedure for the transmission of packets. The Reception procedure specifies the procedure for receiving packets. It invokes the Arbitration procedure for deciding which packet to receive (if any) during the next time slot.

We assume that, new packets at ordinary stations can be transferred to the MAC layer at a maximum rate of one packet per time-slot. Transfers are initiated at the beginning of time-slots. During time-slot t, we say that a packet is *new* if it was generated (and transferred to the station's transmit buffer) during t-1. A packet in the transmission buffer during t[2] is said to be a *signalled packet* if the station signalled (on the control channel) its intention to transmit it during $t-1$.

Let S_i (i=1, 2, ..., N) maintain the following variables for making its transmission/reception decisions.

1. Mini-slot matrix, $H = [h_i]_{N\times 1}$, is an $N \times 1$ array of the N addresses in the first N mini-slots of the current incoming control slot.

2. Recall field matrix, $R = [r_i]_{N\times 1}$, is an $N \times 1$ array for storing the N recall bits of a control slot[3]. $r_i \in \{0, 1\}$.

[2]Recall that throughout this thesis, t is used to denote "time-slot t". Hence, for brevity we can omit the "time-slot" label when referring to the t-th time slot, and refer to it as simply t

[3]The symbol for matrix R is distinguished from the symbol for the i-th receiver (R_i) by

3. Planned reception index, I. At the end of t, $I=j$ if the station will receive the packet on channel λ_j during $t+1$. I is initialised to zero during network start-up.

Procedure Station Transmission (executed by S_i (i=1, 2, ..., N) during every time slot)

CoBegin
if (S_i has a *new* packet) **then**
transmit its destination address on the ith mini-slot of the current control slot, usin
the FT_c of S_i. That packet will be a *signalled* packet during the next time-slot.
if (S_i has a *signalled* packet) **then**
transmit the packet on the current data slot of λ_i (using its FT) .
CoEnd

Comments

1. The reason for transmitting the address of a *new* packet on the station's mini-slot one time-slot prior to the transmission of the actual packet is so that sCA and other stations can be notified of the transmission of the packet one tuning-period in advance. The procedure for receiving mini-slots, and using their information are specified within the 'Station Reception' and the 'sCA Reception' procedures (described next). Note that the precise role and usage of the mini-slots may vary between protocols. Any changes will be described as new protocols are introduced.

2. Each station assumes that the packet it transmitted will be successfully received. Therefore the station can immediately delete the packet from its buffer after transmission. This is why the transmit buffer of a station need a memory capacity for storing just three packets. Stations need not bother with acknowledgements, nor monitor whether its transmission was successful.

3. The **CoBegin** and **CoEnd** constructs indicate that statements within the block are executed in parallel. The conventional { and } constructs still bracket a compound statement, whose constituent statements are sequential executed. For example, in **CoBegin** { a; b; } c; **CoEnd**,

the capitol R followed by a subscript, when referring to a receiver, and by the use of lower case 'r', followed by a subscript when refering to elements of the matrix R.

statements a and b are executed sequentially, and their execution proceeds in parallel with c.

Procedure Station Reception (executed by S_i (i=1, 2, ..., N) during every slot)

```
Begin
    CoBegin
        { receive the addresses in the mini-slots and store them in H;
        receive the values in the recall field storing them in R; }
        if (I ≠ 0) then receive the packet from λ_I ;
    CoEnd ;
I = Arbitration(H,R) ;
End ;
```

Function Arbitration (invoked by S_i (i=1, 2, ..., N) during every slot

Function Arbitration randomly selects for reception one of the packet(s) (destined for S_i) that will arrive during the next time-slot . If none of the packets is destined for S_i, it returns zero, otherwise it returns the index of the channel carrying the selected packet.

Let randomi(m, n, τ=rand()) be the uniform pseudo-random number function [PAWL93a] which generates an integer within [m, n] where m and n are integers s.t. $n \geq m$. randomi uses τ as the random variate, where τ=rand() generates uniform random (real) numbers within [0,1].

Function Station Arbitration(H,R)

```
integer j, k ;
Begin
    k = 0 ;
    if (r_i ≠ 0 ) then return(N+i) ;
    else
      { for j=1 to N do
            if ( h_j== i ) then { k=k+1 ; h_k=j } ;
        if (k==0) then return 0 ;
            else return ( h_randomi(1,k,rand()) ) } ;
End ;
```

4.2.4 Co-operation Between Transmission, Reception, and Arbitration Processes

As one can deduce from the above definitions, the execution of the Transmission procedure is independent of the execution of the Reception procedure. The Reception procedure invokes the Arbitration function.

4.2.5 MAC Protocol of sCA According to the sCA/B Protocol

sCA detects destination conflicts, receives packets that would otherwise be lost, and transmits them so that they arrive as soon as their destinations are free to receive them.

The MAC Protocol of sCA consists of a Transmission procedure called sCA-Transmission, a Reception procedure called sCA-Reception, and a procedure called Plan_Reception which is invoked by sCA-Reception. The sCA-Transmission procedure specifies the procedure for transmitting previously rescued packets from the central buffer of sCA, to their destinations. The sCA-Reception specifies the procedure for identifying packets that need to be rescued, and for their actual reception.

As mentioned, the memory space of sCA is logically partitioned into two fixed size regions, called *tempbuff* and *central buffer*. Let tempbuff be dimensioned to store up to N packets. The remaining memory of sCA is allocated to its central buffer. The central buffer's memory is fully shared by N queues. That is, denoting the length of the queue of packets destined for S_i (including the packet being transmitted during the current slot, if any) by L_i, and denoting the total capacity of the central buffer by L_0, then at any instant in time: $0 \leq L_1 + L_2 + \ldots + L_N \leq L_0$, and $0 \leq L_i \leq L_0$, for any i, i=1,2, ..., N.

Let randomi(m, n, τ) and τ=rand() be the uniform random integer and random real-number functions respectively, as defined before. The sCA station maintains the following variables for making packet reception and transmission decisions:

1. Mini-slot matrix, H, defined in section 4.2.3.
2. Recall field matrix, R defined in section 4.2.3.
3. Planned reception matrix, $P = [p_i]_{N \times 1}$, $i \in 0, 1, \ldots, N$. At the beginning of t, p_i= 1 if during t sCA should receive (into tempbuff) the packet

transmitted by S_i ; p_i, =0 o.w.

4. OccupancY of central buffer, Y. $Y=L_1+L_2+ \ldots +L_N$. $Y=0$ at network initialisation.

5. Occupancy of tempbuff, T, $0 \leq T \leq N$. $T=0$ at network initialisation.

Explanation of the sCA_Transmission Procedure

During each time slot, the sCA transmits[4] R on the recall field of the control slot, and transmits the packets found at the heads of non-empty queues. sCA transmits the packet found at the head of Q_i to S_i on λ_{N+i}, for i=1,2, ..., N. Thus λ_{N+i} is used by sCA to transmit "otherwise lost" packets to S_i. λ_{N+i} will be referred to as the *home channel* of S_i.

The purpose of transmitting R during t is to inform the stations to which sCA plans to transmit a packet during t+1, that they should listen to their home channels. In this way, those stations will be informed one tuning period in advance that they should listen to their home channels, thereby giving them one tuning period to tune their receivers to their home channels to receive the packets from sCA. This role of the recall field has already being defined in the Reception procedure of ordinary stations, see above. The procedure for updating the values of R will be defined in sCA_Reception (later).

The sCA_Transmission procedure can also be defined in pseudo code, as follows.

Procedure sCA-Transmission (Executed by sCA during every slot)
CoBegin
  transmit matrix R on the current recall field using T_c;
  forall T_i i = 1, 2, ..., N **doparallel**
    if ($L_i \geq 1$) **then** { **transmit** the packet from the front of Q_i on λ_{N+i}
    using T_i ; $Y=Y$ - 1 ; }
CoEnd

Recall from section 4.1.3 that T_i and R_i refers to the ith transmitter and receiver of sCA respectively.

Next we will describe the reception procedure of sCA in English. Then we will describe the reception procedure of sCA using pseudo-code. Following

[4]Following established convention, when we speak of "transmitting X" where X is a variable, we mean transmitting the value of that variable.

this we will demonstrate the transmission and reception procedures of stations and sCA using an example.

During every slot, sCA receives a control slot containing the destination addresses of packets that will arrive at sCA during the next time slot. sCA knows the (pseudo) random rule that stations use for selecting the packet to receive. Thus, from these addresses, sCA could identify destination conflicts, and hence could determine which packets it should rescue during the next slot. Simultaneously during the current time-slot, sCA receives packets it previously planned to rescue.

This procedure can also be specified in pseudo code, as follows[5].

Procedure sCA Reception (Executed by sCA during every slot, t=1,2, ...)
CoBegin
  receive the addresses in the incoming control slot into H;
  forall receivers R_i i=1, 2, ..., N **doparallel**
    if (p_i == 1) **then** receive the packet from λ_i using R_i ;
CoEnd
{ j=min(L(0)-Y, T) ; Y=Y + j ; randomly pick up to j packets from those already in tempbuff, transferring them into the central buffer;
  discard remaining T packet(s) in tempbuff, if any ; T=0 ;}
P = Plan_Receptions(H) ;

Procedure sCA Plan_Reception(H) (executed during the "tuning period")
(register) integer j, k ;
integer w ;
conflict analysis matrix, D =$[d_{i,j}]_{N\times N}$, $d_{i,j} \in 0, 1, ..., N$;
conflict count matrix, U =$[u_i]_{N\times N}$, $u_i \in 0, 1, ..., N$;
Begin
  $[u_1, u_2, ..., u_N] = [0, 0, ... , 0]$;
  $[p_1, p_2, ..., p_N] = [0, 0, ... , 0]$;
  $[r_1, r_2, ..., r_N] = [0, 0, ... , 0]$;
  for j=1 to N do
    if ($h_j > 0$) **then**
      { $u_{h_j} = u_{h_j} + 1$; $d_{h_j,u_{h_j}}$ = j ; }
  /* The value of u_i now equals the number of packets from stations arriving to sCA during $t+1$ that are destined for S_i.
  The packets' origin stations are listed in $d_{i,1}$, $d_{i,2}$, ..., d_{i,u_i} */

[5]Recall from section 4.1.3 that T_i and R_i refers to the ith transmitter and receiver of sCA respectively.

```
    for  j=1 to N do
         {  if (u_j > 0) then
           {  if (L_j >1) then {  w = 0 ; }
                else w= randomi(1, u_j, rand() ) ; }
           for  k=1 to u_j do
                if ( k ≠ w) then p_{d_{j,k}}=1 ; T=T+1 ;
         }
    for  j=1 to N do
         if (L_j + p_i >2) then r_j = 1 ;
    return (P) ;
End ;
```

Comments :

1. We assumed that the implementation of rand() in sCA is identical to that in all ordinary stations, and all stations and sCA initialised rand() with the same seed. As shown, rand() is invoked by stations and sCA during every slot, and the (pseudo-random) value it returns is the basis for making reception decisions by ordinary stations (see Function Arbitration). In this way sCA could, from the value returned by rand(), deduce which packets would be received by stations and hence which would not need to be "rescued", i.e. received by sCA for later retransmission (see Procedure Plan_Reception).

2. The increments (in Reception) and decrements (in Transmission) to Y are executed atomically. The meaning is that there are no read nor write operations on the Y variable during the execution of Y=Y+1, apart from the operations for executing this statement. Likewise, these restrictions apply to the execution of Y=Y-1.

3. By letting a destination station randomly select the packet for reception when a conflict occurs, the destination station ensures that it treats all source stations fairly. To illustrate the necessity of the steps in the stations' arbitration function, networks operating according to various simplifications of it are analysed in Appendix D, and are shown to be unfair.

4. By randomly selecting packets for transfer from tempbuff to the central buffer, the sCA ensures that it treats all source stations fairly. To illustrate the reason for this step, a simpler and more intuitive procedure is analysed in Appendix E, and shown to be somewhat unfair.

5. Discarding packets from tempbuff (sCA Reception) ensures that tempbuff has free space for at least N packets at the beginning of every slot. Discarded packets are lost.

4.2.6 Example of Network Operations According to the sCA/B Protocol

Figure 4.7 demonstrates the transmission procedure in a network with N=4 stations. All stations are assumed to be a=5 slots from the star coupler.

Packet Transmission

In Fig. 4.7(a), S_2 wants to send a packet to S_4. During time-slot t it transmits the address of S_4 on its mini-slot. During the next time-slot, S_2 transmits the packet on its data channel, i.e. on λ_2.

These actions are as prescribed by Procedure Station Transmission.

En Route Destination Conflict Resolution

Figure 4.7(b) shows the situation when the control and data packet of S_2 reach sCA (whilst *en-route* to S_4). The time is now the start of $t+a$. By examining the addresses in non-empty mini-slots, sCA finds that both S_2 and S_3 sent a packet to S_4 during the same time-slot. These packets will reach sCA during the next time-slot ($t+a+1$). sCA therefore invokes function Arbitration. Upon determining that the packet of S_2 will be chosen for reception by S_4, sCA knows that the packet sent by S_3 will need to be rescued during the next time-slot. Note that the would-be-successful packet will remain in the optical domain until received by S_4.

These actions are as prescribed by Procedure sCA Reception and Procedure sCA Plan_Reception of the sCA/B protocol.

Figure 4.7c demonstrates the procedure followed by sCA for transmitting the packet that it had rescued. Suppose that during $t+a$, when sCA decided to rescue the packet, it finds that it queue of packets destined for S_4 is empty. Then, during $t+a+1$, sCA can set the 4th recall bit. During $t+a+2$, sCA can transmit the rescued packet to S_4.

These actions are as prescribed by Procedure sCA Transmission of the sCA/B protocol.

Packet Reception

Fig. 4.7(d) shows the situation at S_4 at the start of $t + 2a$. During this time slot, S_4 receives the control slot. First S_4 decodes the 4th recall bit, and finds that it is not set to 1. Then, by decoding addresses in the mini-slots, it deduces that two packets (from S_2, and S_3) intended for itself would arrive during the next time-slot, because its own address (i.e. 4) is in the second and third mini-slot. S_4 then invokes the arbitration function, which randomly picks one of them for reception, in this case, the packet from S_2. Next, S_4 tunes its receiver to λ_2 during the tuning period of $t+2a$ to receive that packet during $t + 2a + 1$.

These actions are as prescribed by Procedure Station Reception of the sCA/B protocol.

Fig. 4.7(e) shows the situation at the start of $t + 2a + 1$. S_4 receives the packet from S_2. S_4 also receives the control slot, and checks the 4th recall bit. Since it finds the 4th recall bit set, it tunes its receiver to λ_{N+4} during the tuning period of $t + 2a+1$ to receive the packet from sCA during $t + 2a + 2$. That packet from S_3 is "otherwise lost".

The packet from S_3 would have been lost if the destination conflict was not resolved en route by sCA. By introducing En Route conflict resolution under the CA/B protocol it is received by S_4 (i.e. its destination) immediately when S_4 is free to receive it.

These actions are as prescribed by Procedure Station Reception of the sCA/B protocol.

4.2.7 The Model and Method Used for the Performance Analysis of sCA/B Networks

The Model

To obtain comparable results, we used the same modelling assumptions as those used in [PAPA92], [CHEN90], [CHEN91], [CHLA91], [HUMB93], [CHEN92] and [CHIP93]. The modelling assumptions are :

A1 The network has N stations.

A2 Station S_i generates new packets following an independent Bernoulli process, with probability p_i that a new packet is generated at a given station during a time-slot.

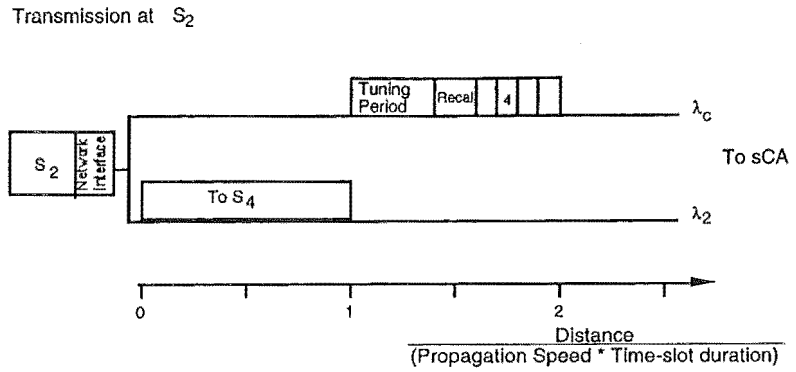


Fig.4.7 (a) Situation at the start of time-slot t +2, showing mini-slot and packet transmitted by S_2 .

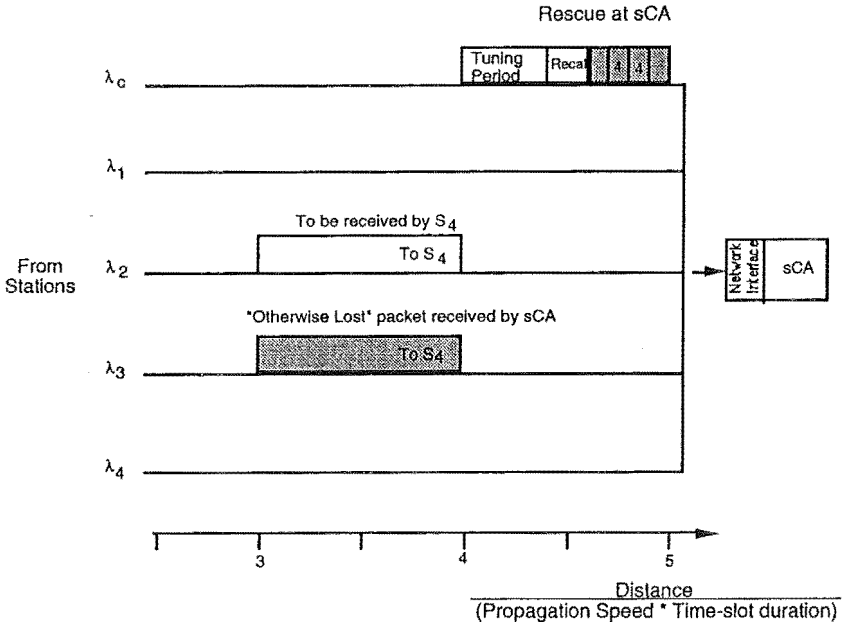


Fig. 4.7 (b) Situation during t+a-1, one time slot prior to destination conflict detection and resolution by sCA

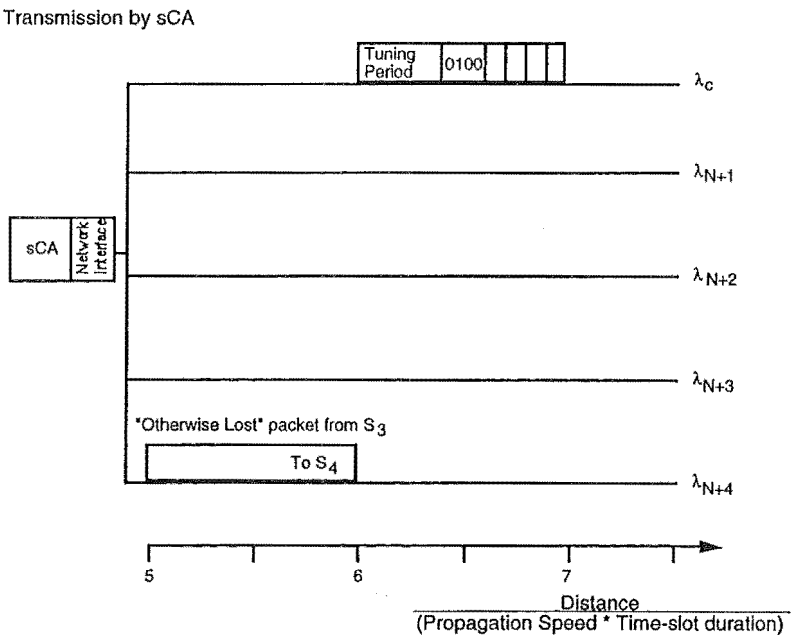


Fig. 4.7 (c) Situation at sCA at the end of t+a+2, after sCA transmited the "otherwise lost" packet it rescued to its destination.

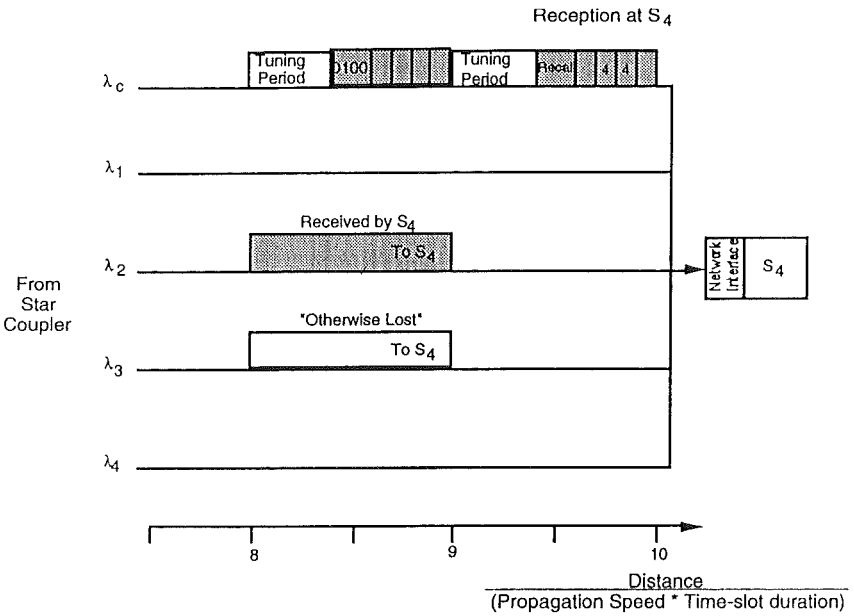
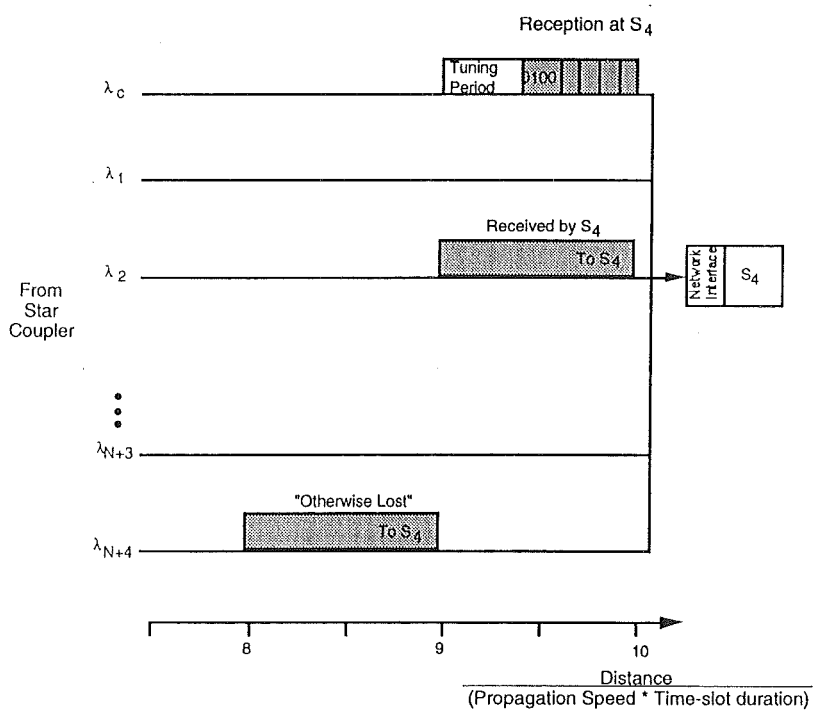


Fig. 4.7 (d) Mini-slots reception at the start of $t+2a$ and packet to be received during $t+2a+1$ (indicated by shading) by S_4 .



e) Situation at S_4 one time-slot later. During $t+2a+1$ S_4 receives the packet from S_2. During $t+2a+2$ S_4 receives the "otherwise lost" packet from S_3 on λ_8.

Figure 4.7: Time-Frequency diagram showing the procedure followed by stations for transmitting packets (a), the detection and resolution of destination conflicts by sCA (b) and (c), and the procedure followed by the stations for receiving packets (d) and (e), in an sCA/B network. a=5, N=4. Signals to be received are indicated by shading.

A3 Stations are assumed to have a uniform new packet generation rate p. This assumption implies that $p_1 = p_2 = ... = p_N = p$. p will be referred to as the normalised load.

A4 Station S_i is a_i slots from the star coupler (and hence sCA). The one way propagation delay is $2a$ slots when stations are equidistant from sCA.

A5 A uniform reference model is assumed. This implies that the destination of a packet is chosen from any of the N-1 other stations with equal probability $1/(N\text{-}1)$.

A6 Every station has a transmit buffer with a capacity for storing B packets.

We follow conventional notation by assuming that every station has been provided with memory for storing up to B packets. However in sCA-STAR this buffer memory located in sCA [6]. At sCA, memory for storing up to N packets is assigned to tempbuff. Hence, sCA's central buffer has enough memory to store Q_0=NB-N packets.

The goodness of fit of the above model depends on the specific network application. If each station is a multi-user multitasking workstation, then the new packets generated by the processes executed by each user may be statistically multiplexed for transmission. Whilst the generation times of new packets by a single user process may be correlated, the multiplexed new packet stream is likely to exhibit lower correlation, provided that user processes are largely independent, and the number of user processes generating new packets is not too small. In the case of multi-user multitasking stations, the Bernoulli packet generation process assumption, and the uniform reference model, may therefore be a reasonable approximation.

The generation times of new packets by a single station, depending in part on the type of the network application(s) executing on the station, their requirements (e.g. variable bit rate versus constant bit rate), the compression method used, and the size of packets. In addition to correlated generation times, each packet burst is likely to have the same station as its destination. For example, a series of packets which contain a file to be transmitted to a station would all have that station as their destination. Also, in many applications, the destinations of packets generated by a station may not be purely random, but can be restricted to subsets of network stations. This property will be refereed to as the *locality of reference* .

[6]Strictly, buffer memory for B-3 packets per station are allocated to sCA, assuming that memory for three packets is still needed at each station.

We will concentrate our analysis assuming the above model hereafter. The influences of assumptions A2, A3, and A5 were studied by considering sCA-STAR networks modelled using a Markov Modulated Bernoulli Process new packet generation model, an asymmetric load model, and a non-uniform destination reference model respectively. These experiments, and their results are contained in Appendix D.

Performance Measures

The main performance measures considered are :

- *normalised throughput* defined as the mean number of successful packet receptions per station per time-slot; and
- *average packet delay* defined as the average of time intervals from when a new packet is generated (i.e. transferred to the MAC layer at the source station) to when it is successfully received by its destination.
- We will also consider the *mean excess delay of packets generated at station* S_i. The *excess delay* of a packet from S_i represents the delay experienced in excess of the packet's *ideal delay* (shortest possible delay). The ideal delay of a packet equals $a_s + a_d + 1$, where a_s and a_d are the propagation delays from the source to the star coupler and from the star coupler to the destination station respectively. Hence the excess delay of a packet from S_i equals its total delay minus $(a_i + a_d + 1)$.

Method of Analysis

The steady-state performance characteristics of sCA/B networks were analysed by means of quantitative stochastic simulation. All simulators were written and executed using AKAROA, an object-oriented simulation package developed by us for controlling the precision of steady-state estimates and for the automated execution of quantitative simulations in parallel. AKAROA transparently transforms a sequential simulation program into one for parallel execution on a network of workstations. The Spectral Analysis in Parallel Time Streams method, a parallel version of the method given in [HEID81], is used for sequential simulation output data analysis. The procedure given in [PAWL90] was used for detecting the onset of steady state of each process analysed. Simulation runs were stopped when the steady-state estimates of all performance measures obtained the relative precision (relative width of the confidence interval) of less than 5%, at the confidence level of 95%.

The AKAROA network simulation package and its use in network modelling and performance evaluation, is discussed in some detail in a separate technical report [YAU96a].

4.2.8 Results

First, consider sCA-STAR networks operating under the sCA/B protocol. Each network has N=10 stations, where all stations are a=5 slots from sCA.

Effect of central buffer size on throughput/delay characteristics Fig. 4.8 shows the (normalised) throughput as a function of offered load for B=10, 15, 20, 30, 40. It can be seen that the throughput performance of sCA/B CA-Star networks is almost optimal. That is, their throughput almost equals the offered traffic except at the highest level of traffic (p=1), where throughput reached 96.13% when B=10, and approximately 98.52% when B=40.

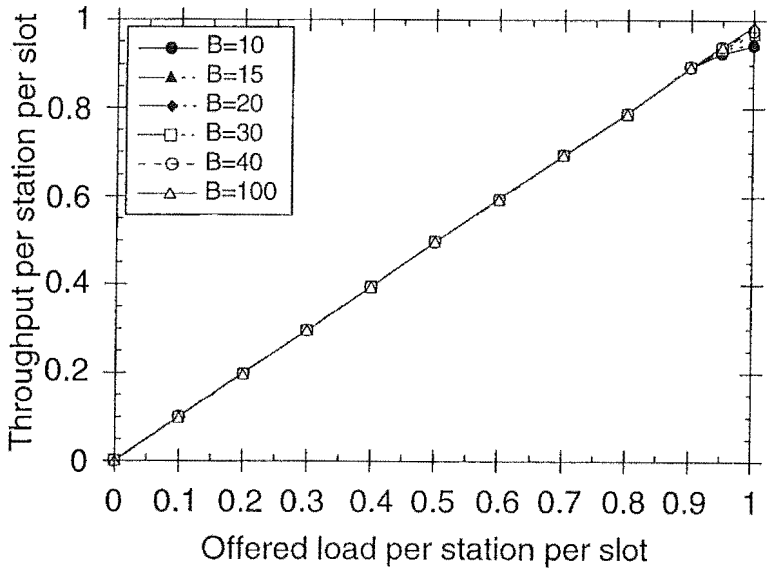


Figure 4.8: Throughput of sCA/B versus normalised load, for varying buffer memory capacity of sCA. Relative precision: ≤5%.

In Fig. 4.9 the average packet delay is plotted as a function of the offered load. Provided that the offered load is below 90%, the mean delay experienced by packets is close to the minimum of one source-to-destination propagation delay plus one slot (11 slots), for all values of B considered. At p=1, the mean delay has increased to near B+2a+1.

We can conclude that for an N=10 sCA-STAR operating according to

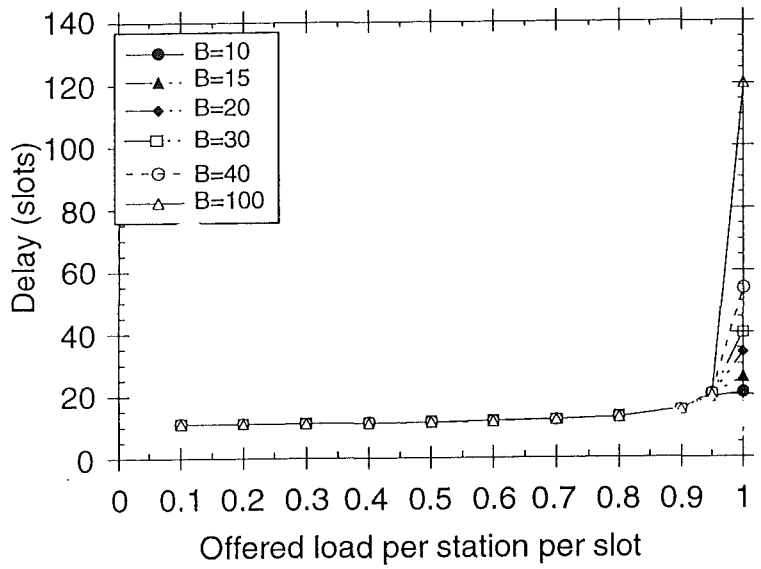


Figure 4.9: Mean packet delay of sCA/B versus normalised load, for varying buffer memory capacity of sCA. Relative precision: $\leq$5%.

sCA/B, almost optimal throughput and delay is obtained already when B=10, provided that the operating load p is below 90%. When p=1, increasing B above 10 improves throughput somewhat, but also increases the average packet delay.

Impact of increasing network size Now, let us investigate whether B would need to be increased as the network size grows by taking a B=25 and a=5 sCA/B network as a benchmark. The effects of increasing network size were studied by considering the throughput and mean packet delay when N=3, 5, 10, 20, 30, 40, and 100. Results are graphed in Figs. 4.10 and 4.11. One can see from Fig. 4.10 that increasing the number of stations in the sCA/B network from 10 to 100 does not affect its efficiency. Increasing network size from 3 to 5 stations increases the average packet delay somewhat at medium traffic, see Fig. 4.11, but further increases has little effect. These results demonstrate that sCA/B remains a good solution as network size increases, *even when B remains constant.*

Fairness For WDM Star networks, it is interesting to see if all stations are treated fairly in terms of throughput and mean packet delay, irrespective of their position from the star coupler.

The fairness of sCA/B networks was investigated by studying a network with N=10 stations, where station S_i is located a_i slots from sCA, where

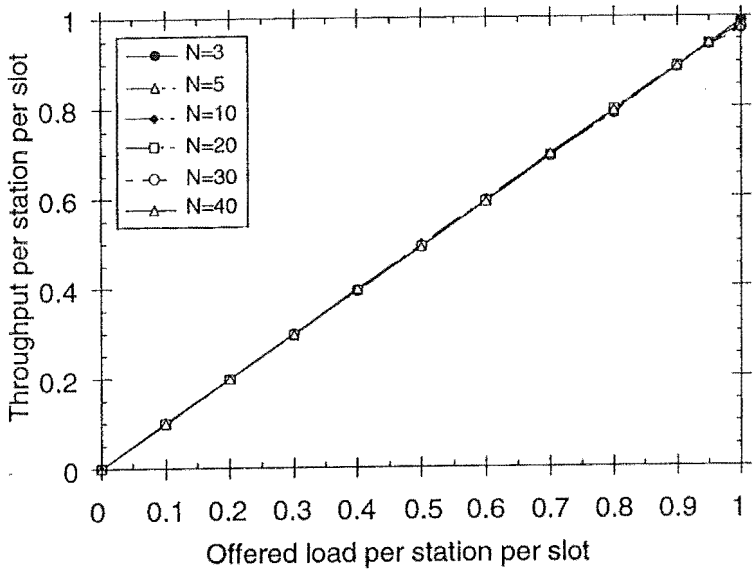


Figure 4.10: Throughput of the sCA/B network versus normalised load, for varying number of stations. Relative precision: $\leq$5%.

a_i=$2i$, $1 \leq i \leq 10$. B was assumed to be 10. Estimates of the mean excess packet delay are plotted as a function of station index for various traffic levels in Fig. 4.12. The throughput per station per slot is plotted as a function of station index for various traffic levels in Fig. 4.13. From Figs. 4.12 and 4.13, it appears that the throughput and mean excess delay is independent of station index, suggesting that the sCA/B protocol treats stations equally. Also, mean excess delay is very low until near p=1.

Results for the same network, but with B=20 are summarised in Figs. 4.14 and 4.15. An examination of the throughput and mean packet delay at different stations in this case suggests that the sCA/B protocol treats stations fairly, regardless of their positions, and regardless of the levels of traffic load, and buffer capacity.

Unfairness could occur if sCA followed a slightly relaxed (but intuitively simpler) protocol (see Appendix D).

Heavy traffic studies Given that the throughput of sCA/B with B=10, 20, 30, 40, were almost identical and indistinguishable from ideal (i.e. throughput almost equals offered load) except at the highest possible traffic level, an investigation of throughput/delay for a wider range of B, when p=1 is a meaningful next step. Results for a sCA/B network with N=10 stations, and

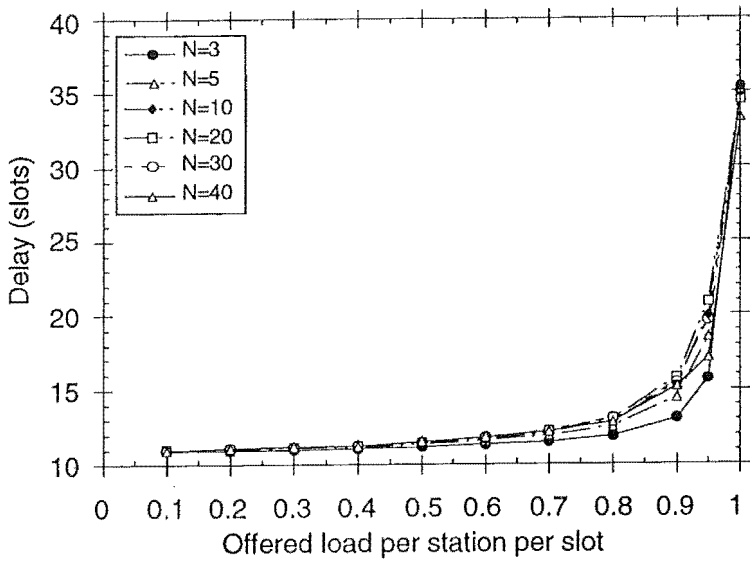


Figure 4.11: Mean packet delay of the sCA/B network versus normalised load, for varying number of stations. Relative precision: $\leq$5%.

a=5, are reported in Table 4.1. The first column gives the capacity of the central buffer memory, divided by the number of stations (B_{cb}).

$$B_{cb} = B - 1 = Q_0/N \tag{4.1}$$

The second column gives the values of throughput followed by their final confidence intervals at the 0.95 confidence level. The mean excess delay estimates are presented in the same format in column three.

Table 4.1 shows that the throughput of the network increases from 81% to 95% as B_{cb} increases from 1 to 10. Increasing B_{cb} above B_{cb}=10 yields diminishing returns.

Excess delay increases with increasing B_{cb}. Under maximum load (p=1), the network transmits packets at the maximum rate. Throughput/delay performance would be ideal only if the reception capacity of the network was always productively employed. sCA helps achieve this by transforming batch arrivals (destination conflicts) into a stream of single packet arrivals. An examination of Table 4.1 suggests that about B=10 gives sCA sufficient buffer capacity for its task. Adding more buffer memory to sCA increases the total number of packets that can be stored within the sCA-Star system, but it does not change the total reception capacity of the system. Hence MAC delay increases.

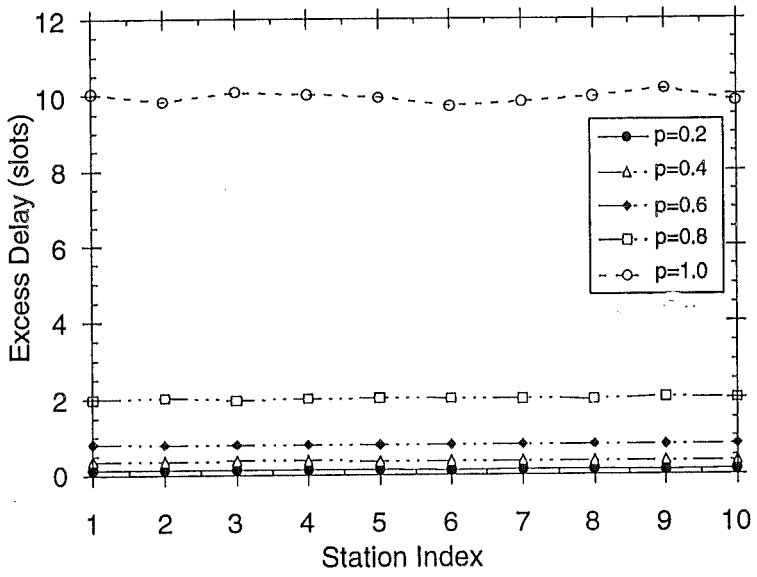


Figure 4.12: Mean excess packet delay as a function of station index, in an sCA/B network with B=10, N=10, a_i=2i. Relative precision: $\leq$5%.

Probability of packet loss Knowing the level of traffic that an sCA/B network can support, whilst providing a given quality of service (in terms of maximum probability of packet loss and mean packet delay), is helpful for network management.

The probability of packet loss as a function of the normalised offered traffic, p, was investigated by considering a 10 station sCA/B network with B=10, and a_i=5, i=1, 2, ..., 10. Results are contained in Table 4.2. One can see that the probability of packet loss is about 0.04 when p=1, but diminishes quickly as p decreases.

This "/B" type of protocol may serve network applications where some packet loss is acceptable, provided that the probability of packet loss is below a specified level. Due to the uniform reference and the symmetric source assumptions, the value of p also equals the mean number of new packets destined to a given station that are generated during a time-slot. Consequently the above results suggests that the probability of packet loss diminishes rapidly as the mean arrival rate of packets for a station (which equals p) is reduced from 100% of the reception capacity of that station. In connection oriented packet switched networks, the mean arrival rate of packets for a station, and hence the probability of packet loss, can be kept below a specified level by the use of a connection acceptance function by stations when deciding whether to accept new connection requests.

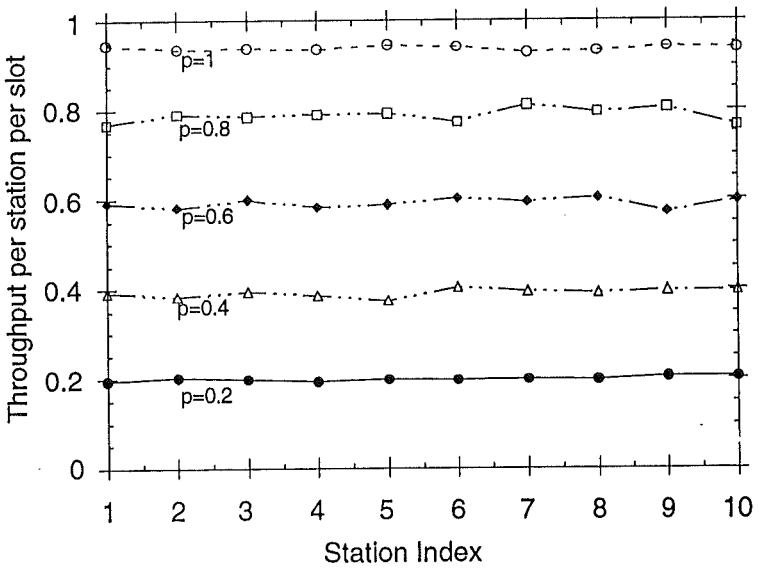


Figure 4.13: Throughput as a function of station index, in an sCA/B network with B=10, N=10, a_i=$2i$. Relative precision: $\leq$5%.

These protocols may suit applications where the major bandwidth consumers such as HDTV, video-on-demand, voice, and teleconferencing applications can tolerate some packet loss. Loss averse users such as distributed databases can also be accommodated by "/B" protocols such as sCA/B, if the probability of packet loss is not too high. For these applications, the losses of packets must be detected by higher protocol layers, and resolved by retransmitting the protocol data unit (PDU) containing the lost packet.

Thus sCA/B may not be suitable for network applications where the *majority* of bandwidth users are loss averse. Next we introduce a variation of sCA/B, which could ensure that packets are never lost.

4.3 The sCA/R Protocol

Under the sCA/B protocol, packets may be lost (discarded from tempbuff) if the central buffer is full. Thus for applications where any losses of packets is unacceptable, lost packets must be retransmitted by retransmitting their corresponding PDU.

To serve networks where the majority of applications cannot accept any packet loss, this section introduces a variant of the sCA/B protocol which

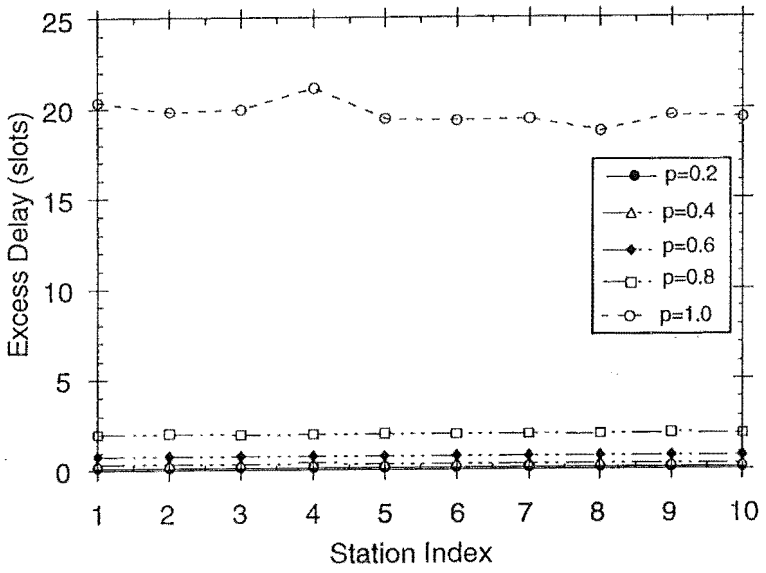


Figure 4.14: Mean excess packet delay as a function of station index, in an sCA/B network with B=20, N=10, a_i=2i. Relative precision: $\leq$5%.

provides guaranteed delivery of packets through the use of a procedure called *reflection*. Following the convention for naming protocols introduced in Chapter 1, this protocol will be named sCA/R.

4.3.1 Reflection

In sCA/B networks, packets in the tempbuff region of the buffer of sCA could not be "transferred" into the central buffer region when it is full, and are consequently lost. To guarantee successful delivery of a packet once it has been transmitted by its origin station, we introduce a simple deflective-routing/back-pressure procedure named *Reflection*. This procedure consists of 2 steps:

1. If there are packets in tempbuff which cannot be transferred into the central buffer of sCA, then transmit these packets to idle stations instead of loosing them. A station is said to be *idle* during a given time slot, if during that time slot no packets are destined for it from stations or from the central buffer of sCA.

   Packets vectored to idle stations are called *reflected packets*. Idle stations selected as destinations of reflected packets are called *surrogate destinations*.

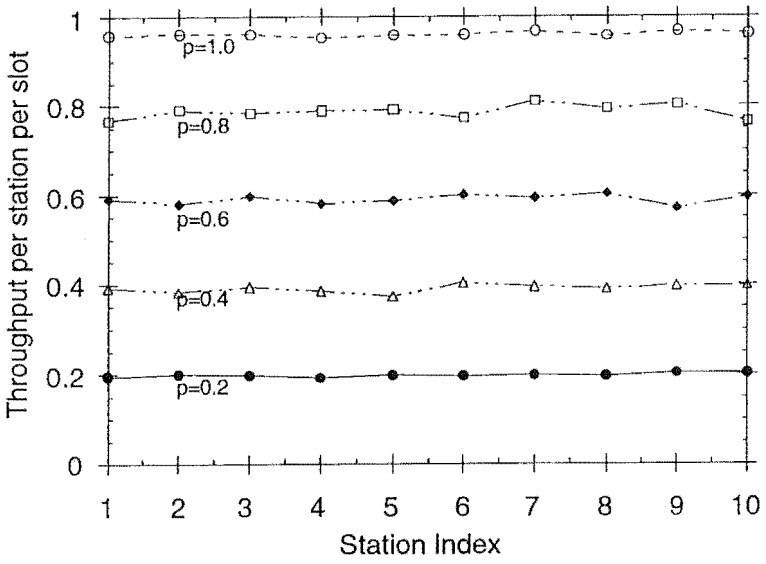


Figure 4.15: Throughput as a function of station index, in an sCA/B network with B=20, N=10, a_i=2i. Relative precision: $\leq$5%.

2. During t, if an idle station receives a packet that is not destined for itself, it retransmits that packet to its destination in the usual way. That is, during t+1, it treats the packet as one that has just being locally generated.

   During t+1, the station also blocks the submission of its own new packets, if any.

The pseudo-code of the sCA/R protocol, incorporating the above Reflection procedure, is specified in Appendix E

4.3.2 Proof that Reflection is Correct

To proof that Reflection eliminates packet losses[7] regardless of the pattern or the intensity of the offered traffic, it suffices to show that at any time-slot t the number of stations that could serve as surrogates equals or exceeds the number of packets that need to be reflected during t. Let s denote the number of stations that could serve as surrogates. A station can serve as a surrogate to a packet reflected from sCA during t, if the station would be idle (no packet destined for it) when the reflected packet arrives.

[7]safety property

B_{cb}	Through put (sCA/B)		Mean Excess Delay (sCA/B)	
1.000	0.8099	(0.8045, 0.8153)	1.040	(1.033, 1.047)
2.000	0.8679	(0.8614, 0.8744)	2.045	(2.029, 2.061)
3.000	0.8861	(0.8777, 0.8944)	3.042	(3.021, 3.064)
4.000	0.9174	(0.9096, 0.9252)	4.036	(4.007, 4.065)
5.000	0.9258	(0.9167, 0.9349)	5.032	(4.985, 5.080)
6.000	0.9310	(0.9223, 0.9397)	6.026	(5.991, 6.061)
7.000	0.9400	(0.9321, 0.9478)	7.016	(6.974, 7.058)
8.000	0.9455	(0.9381, 0.9529)	8.002	(7.955, 8.048)
9.000	0.9486	(0.9412, 0.9560)	8.989	(8.937, 9.041)
10.00	0.9511	(0.9440, 0.9582)	9.976	(9.899, 10.05)
11.00	0.9549	(0.9482, 0.9616)	10.96	(10.87, 11.05)
12.00	0.9479	(0.9396, 0.9561)	11.95	(11.85, 12.04)
13.00	0.9493	(0.9413, 0.9572)	12.93	(12.83, 13.02)
14.00	0.9522	(0.9445, 0.9599)	13.90	(13.79, 14.02)
20.00	0.9636	(0.9560, 0.9711)	19.77	(19.60, 19.93)
30.00	0.9781	(0.9694, 0.9868)	29.38	(29.20, 29.55)
40.00	0.9781	(0.9694, 0.9868)	38.80	(38.51, 39.09)
50.00	0.9781	(0.9694, 0.9868)	47.71	(47.27, 48.15)
60.00	0.9781	(0.9694, 0.9868)	56.11	(55.72, 56.51)

Table 4.1: Throughput and mean packet delay of sCA/B networks as a function of B_{cb}, where the memory capacity of the central buffer of sCA equals $B_{cb}N$.

As assumed tempbuff has storage capacity for $2N$ packets. During any slot, at most N packets need to be rescued by sCA, taking into account all possible destination conflicts of packets transmitted by all stations and by sCA. The maximum occupancy of tempbuff at the beginning of a time-slot should be N, to be sure that all packets that need rescuing can be received into tempbuff.

Let R_{max} be the maximum number of packets that need to be reflected during a slot t. By the above constraints, R_{max} must be equal to the occupancy of tempbuff at the beginning of that time-slot, plus the number of packets that will be received during t, minus N, minus the number of packets that could be transferred from tempbuff to the central buffer at the end of t. This can be expressed analytically as

(c2) $\quad R_{max} = \max(0, T^{\oplus} - N - j^{\oplus} - b)$

where

p	P (packet loss)		Thr oughput		Mean Excess Delay	
1.000	0.04207	(0.04043, 0.04370)	0.9452	(0.9337, 0.9567)	9.083	(8.666, 9.499)
0.9900	0.03413	(0.03255, 0.03572)	0.9623	(0.9512, 0.9735)	7.875	(7.512, 8.238)
0.9800	0.02534	(0.02424, 0.02644)	0.9490	(0.9384, 0.9597)	8.965	(8.671, 9.258)
0.9700	0.01775	(0.01710, 0.01839)	0.9401	(0.9327, 0.9474)	8.459	(8.094, 8.824)
0.9600	0.01071	(0.01032, 0.01109)	0.9485	(0.9401, 0.9570)	8.028	(7.666, 8.390)
0.9500	0.005078	(0.004907, 0.005249)	0.9465	(0.9315, 0.9615)	7.777	(7.486, 8.068)
0.9400	0.002081	(0.0019906, 0.0021719)	0.9336	(0.9219, 0.9452)	7.630	(7.318, 7.941)
0.9300	0.0006448	(0.0006148, 0.0006748)	0.9244	(0.9130, 0.9359)	5.610	(5.358, 5.862)
0.9200	0.0001543	(0.0001485, 0.0001602)	0.9094	(0.8957, 0.9231)	5.054	(4.808, 5.299)
0.9100	3.0502e-05	(2.9003e-05, 3.2001e-05)	0.8960	(0.8818, 0.9102)	5.084	(4.836, 5.331)

Table 4.2: Probability of packet loss and mean packet delay in sCA/B networks as a function of p, the normalised offered load. B=10 , N=10, a=5.

- $T^{\oplus}$ is value of T at the start of t, as defined above. $T^{\oplus}$ equals the number of packets in tempbuff at the start of t *plus* the number of packets that will be received during t.
- $Y^{\oplus}$ is the central buffer occupancy at the beginning of t
- $j^{\oplus}$ is the number of free spaces in the central buffer[8] at the beginning of t.

$$j^{\oplus} = L_0 - Y^{\oplus} \tag{4.2}$$

and

- b is number of packets that will be transmitted from the central buffer to their destinations during t.

Define s as the number of stations that can serve as surrogate stations. Surrogate stations would not have any packets destined for them during t (neither from other stations nor from sCA).

[8]Recall that L_0 denotes the total capacity of the central buffer

Assume that at the beginning of t, there are at most N packets in tempbuff. If P is the number of packets in the incoming data slots during t that need buffering by sCA ($0 \leq P \leq N$), then

$$T^{\oplus} \leq N + P, \tag{4.3}$$

implying that

$$R_{max} \leq N + P - j^{\oplus} - b - N = P - j^{\oplus} - b \tag{4.4}$$

In the worst case the central buffer is full at the beginning of t, so $j^{\oplus}$ equals zero. Assuming $j^{\oplus}$ equals zero we get

$$R_{max} \leq P - b \tag{4.5}$$

Let k of the P packets be involved in a destination conflict with packets transmitted by sCA, $0 \leq k \leq P$, and $k \leq b$. Then P-k packets arriving to sCA during t need buffering because of destination conflicts with packets transmitted by other stations (i.e. not by sCA). Thus there are at least P-k stations that will not have any packets destined for them that were transmitted from ordinary stations. Also $b - k$ of the packets transmitted by sCA would not cause any destination conflicts. Hence, at most $b - k$ of the P-k stations would have a packet to receive. The number of stations that will not have a packet destined for them (neither from ordinary stations nor from sCA) is therefore

$$s \geq P - k - (b - k) \tag{4.6}$$

Thus from Eqns. 4.5 and 4.6 we obtain

$$R_{max} \leq s \tag{4.7}$$

as required.

We can conclude that sCA could transmit (reflect) at least R_{max} packets from tempbuff to surrogate stations during t. Their successful reception by those stations is also guaranteed. This completes the proof since it implies that the occupancy of tempbuff at the beginning of a time-slot can never exceed N, hence at least N new packets can be admitted. Also, having received a reflected packet, a station would reflect (re-transmit) it to its destination. From above, it will not be lost in the process.

4.3.3 Results of Performance Studies

Effect of central buffer size on throughput/delay Fig. 4.16 shows the (normalised) throughput obtained by sCA/R networks as a function of offered load for B=10, 15, 20, 30, 40. N is assumed to be 10 stations, and a is 5. All values were estimated using AKAROA to within $\leq$5% of the population parameter at the 95% level of confidence. The same model of the sCA/B study was assumed, see section 4.2.7 on page 86.

One can observe that sCA/R provides almost optimal throughput. That is, the throughput of the sCA/R networks almost equals the offered traffic except at the highest load level, p=1, where their throughput reached about 96% when B=10, and approximately 98% when B=40.

Estimates of the mean packet delay of sCA/R networks are plotted as a function of the offered load in Fig. 4.17. The results show that the mean delay experienced by packets is close to the minimum (11 slots), for all values of B considered, provided that the offered traffic is below 90%. Above p=0.9, the mean delay increases to near B+2a+1.

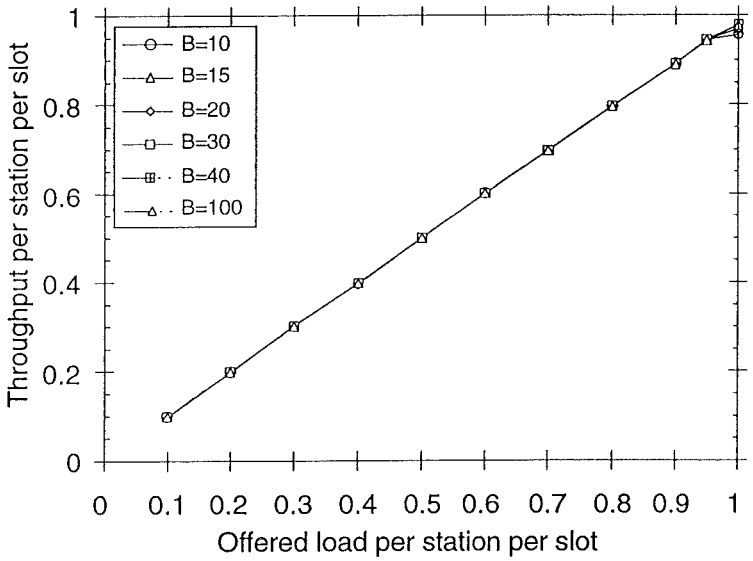


Figure 4.16: Throughput of sCA/R versus load, for varying buffer sizes. N=10, a=5. Relative precision: $\leq$5%.

Impact of increasing network size The effects of increasing network size when B is fixed were explored by considering sCA/R networks with N=3, 5, 10, 20, and 40 stations; a=5, and B=25. Results are graphed in Fig. 4.18 and Fig. 4.19.

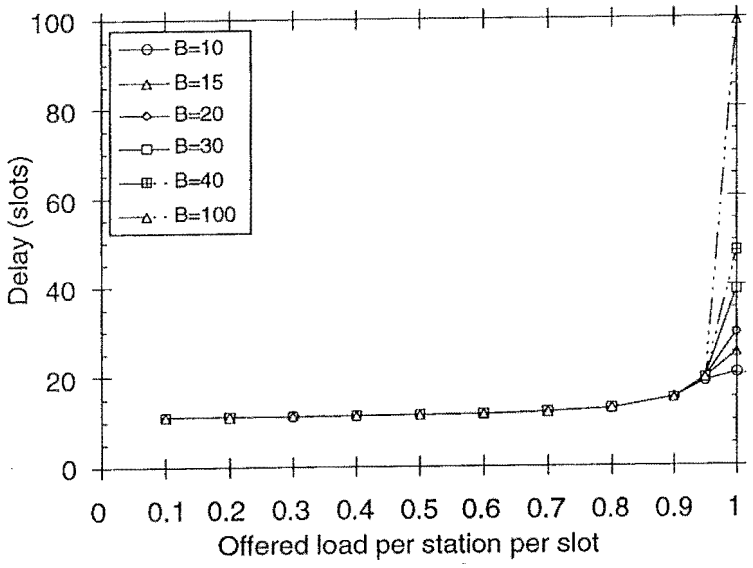


Figure 4.17: Mean packet delay of sCA/R versus load, for varying buffer sizes. N=10, a=5. Relative precision: $\leq$5%.

One can see that increasing the number of stations in the sCA/R network from 10 to 40 does not affect its efficiency. Increasing network size from 3 to 5 stations increases the average packet delay somewhat at medium traffic, see Fig. 4.11, but further increases have little effect. These results demonstrate that the sCA/R protocol remains a good solution as network size increases, *even when B remains constant.*

Heavy traffic behaviour Given that the throughput of sCA/R networks with B=10, 20, 30, 40, almost equalled the offered load, except at the highest possible traffic level, an investigation of their throughput and delay when p=1, for a wider range of B, is a meaningful next step.

The results for an sCA/R network with p=1, N=10, and a=5 are reported in Table 4.20. The first column gives the values of the memory available to the central buffer per station (B_{cb}). The second contains the values of throughput, followed by their final confidence intervals at the 95% level of confidence. The mean excess delay estimates are presented in the same format in column three.

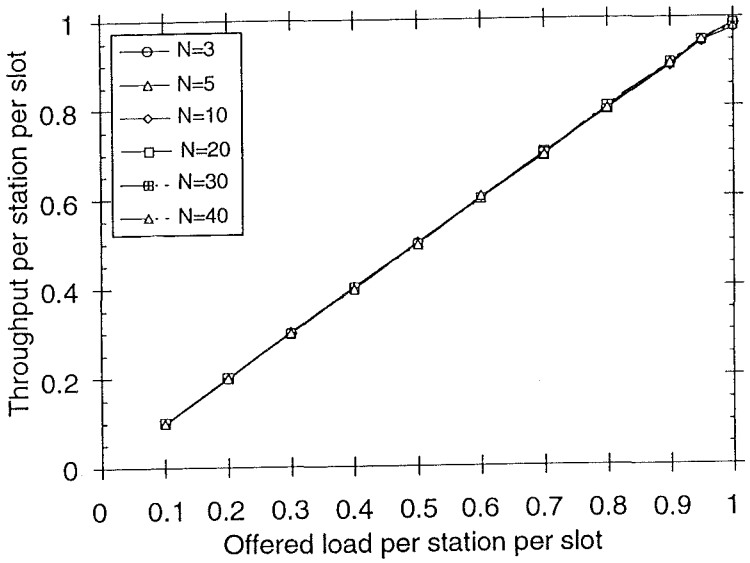


Figure 4.18: Throughput of sCA/R versus load, for varying number of stations. B=25, a=5. Relative precision: $\leq$5%.

An examination of Table 4.20 shows that the normalised throughput of the network increased from 86.6% when B_{cb} is 2, to 97.2% when B_{cb} is 10. Increasing B_{cb} above B_{cb}=10 yields diminishing returns.

The mean excess delay increases with increasing B_{cb} as well. The positive relationship between B_{cb} and the average packet delay can be due to the fact that at maximum offered load, increasing B_{cb} increases the number of packets that could be accepted into the network from stations' upper (LLC) layers. Yet the network's reception capability is saturated, so a larger number of admitted packets mean a longer waiting time before packets are delivered, as expected from Little's result.

However, we see from the first row that delay again increases as B *decreases* from 4 to 2.4. This anomaly can be intuitively explained for sCA/R, since the probability that central buffer is full increases when B is reduced. Mean excess delay is increased because excess packets are reflected from sCA when its central buffer is full, and each time a packet is reflected its delay increases by $2a$+3 time-slots.

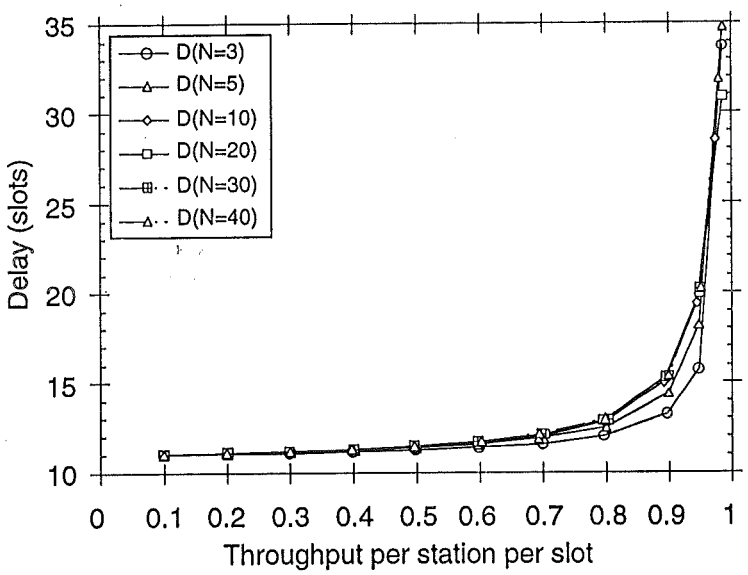


Figure 4.19: Average packet delay of sCA/R versus load, for varying number of stations. B=25, a=5. Relative precision: $\leq$5%.

4.4 Chapter Conclusions

The sCA-STAR architecture is the implementation of en route conflict resolution presented in this chapter. The performance of sCA-STAR operating according to the sCA/B and sCA/R media access control protocols were analysed using the AKAROA distributed simulation and runlength control software. The main performance measures studied were: network throughput, mean packet delay, and probability of packet losses.

Both sCA/B and sCA/R protocols shows very good performance. In the networks where $B \geq 10$, the capacity of these networks is very close to 100%, and the average packet delay is very close to the minimum, provided that the offered load was below 90%.

sCA-STAR enjoys high throughput, since the "otherwise lost" packets are rescued by sCA, and transmitted to their destinations from sCA as soon as their destinations are free to receive them. In sCA-STAR, packets would not incur addition propagation delay even if they are involved in a conflict, and would be "otherwise lost". This is because "otherwise lost" packets are rescued en route at the entrance to the star coupler, and would be transmitted from that point. Packets are buffered by sCA only if they need to wait for one or more time-slots until their destination is free (i.e. only if they would

B_{cb}	Through put (sCA/R)	xD (sCA/R)
1.000	0.7951 (0.7874, 0.8028)	4.670 (4.624, 4.716)
2.000	0.8666 (0.8585, 0.8747)	4.525 (4.485, 4.566)
3.000	0.8910 (0.8840, 0.8979)	4.939 (4.905, 4.972)
4.000	0.9107 (0.9019, 0.9195)	5.602 (5.566, 5.638)
5.000	0.9305 (0.9241, 0.9370)	6.355 (6.293, 6.418)
6.000	0.9379 (0.9291, 0.9467)	7.182 (7.142, 7.223)
7.000	0.9442 (0.9356, 0.9527)	8.125 (8.087, 8.162)
8.000	0.9516 (0.9443, 0.9589)	9.001 (8.912, 9.089)
9.000	0.9602 (0.9524, 0.9679)	9.912 (9.843, 9.980)
10.00	0.9720 (0.9647, 0.9792)	10.86 (10.79, 10.93)
11.00	0.9712 (0.9639, 0.9786)	11.74 (11.63, 11.86)
12.00	0.9688 (0.9604, 0.9773)	12.79 (12.70, 12.88)
13.00	0.9589 (0.9520, 0.9658)	13.69 (13.57, 13.80)
14.00	0.9612 (0.9534, 0.9690)	14.59 (14.47, 14.71)
20.00	0.9752 (0.9662, 0.9841)	20.05 (19.88, 20.21)
30.00	0.9789 (0.9732, 0.9846)	24.93 (24.69, 25.18)
40.00	0.9789 (0.9732, 0.9846)	34.98 (34.66, 35.30)
50.00	0.9789 (0.9732, 0.9846)	45.11 (44.91, 45.31)
60.00	0.9789 (0.9732, 0.9846)	51.09 (50.78, 51.40)

Figure 4.20: Throughput and mean excess packet delay of sCA/R networks as a function of B_{cb}. p=1, a=5.

otherwise be lost). Hence the delay of one time slot for the O-E conversion of a packet that need to be rescued (received) by sCA can be considered as desirable.

The performance congruency of sCA/R with sCA/B suggests that the Reflection procedure carries negligible performance overhead. This was intuitively expected, since from Tables 4.2 one can determine that the probability that sCA is depleted of buffer memory is very low, provided $B \geq 10$. Hence Reflection is rarely invoked in sCA/R networks.

sCA/B is the most appropriate protocol for sCA-STARs operating in a connection oriented mode, if the major bandwidth consumers in the network can tolerate some packet losses (due to occasional depletion of buffer memory at sCA). The probability of packet loss and the mean packet delay can be kept below a specified level, by applying an appropriate connection acceptance control function at the destinations of requested virtual circuits, during connection setup. Using sCA-STAR with the sCA/B protocol guarantees that the delay of a packet is bounded by $BN+1$ time slots. sCA/B preserves the order of packets within all connections.

The sCA/R protocol is more advanced of the two sCA-STAR protocols studied. sCA/R guarantees that packets are never lost throughout the use of a deflection routing procedure called Reflection. Due to reflections, packets may not always be delivered in the order that they were transmitted. The use of sequence numbers in packets may therefore be necessary, as in traditional packet switched networks.

By resolving conflicts at CA where data signals are still Space Division Multiplexed (prior to entering the star coupler), *at most one* packet per station needs rescuing per time-slot. Hence, CA-Star differs from "receiver-replication" based solutions in that neither multiple tuneable-filters/delay-lines nor multiple receivers are needed per station. However, it must be noted that the sCA central arbiter design assumes that its memory supports up to N concurrent read and writes, provided they are on distinct locations. This requirement implies that the memory must support complex access modes. Alternatively, a memory design that supports only one packet read and write operation at a time can be used, provided that up to N read and write accesses can be served per time-slot. Since the transmit buffer memory of the network interface of ordinary stations need to support at most one packet read access per time-slot, this implies that the memory of sCA has to operate N times faster than that of ordinary stations.

Chapter 5

optCA-STAR Networks

One of the assumptions made when designing sCA-STAR declares that the memory of sCA can service N packet read and write access requests per time-slot, provided that they are on distinct locations (addresses). Thus the bandwidth of sCA's common memory has to be N times that of the memory bandwidth of ordinary stations, where N is the number of stations in the network. If the number of stations in the network is large, this assumption may be too optimistic. Supporting N simultaneous read/write accesses to the common memory would require a complex busing structure and memory organisation. Alternatively, if the requests are serviced in sequence, then the maximum time needed by sCA for servicing accesses to its memory during one time-slot increases as N increases. The transmission time of a packet, and hence the duration of a time-slot is unchanged by increases to the value of N, provided that the number of mini-slots per control slot could be transmitted within one time slot. Consequently, the memory operations of sCA may develop into the network's (electronic) bottleneck, as new stations are added to the network.

So that memory access operations of the CA need only proceed at the same speed as the operations at ordinary stations - independent of N; and to reduce the complexity of the busing structure of CA to the level of ordinary stations (again regardless of N), a simplified central arbiter design, called optCA, is introduced in this chapter.

Unlike sCA, the optCA station was designed to be suitable for using either **opt**ical or electronic buffers, for storing packets that it has rescued. Hence it is named **opt**CA. The name "optCA" is a label for the central arbiter design considered, and does not imply that optCA is restricted to using optical buffers. As mentioned, electronic buffering is also an interesting option because (unlike sCA) the memory access operations of the optCA buffers need

only proceed at the same speed as the operations at ordinary stations, and that the delay of one time slot during which the optical to electronic conversion of "otherwise lost" packets takes place is *desirable*. This chapter develops the optCA-STAR architecture and proposes a protocol for its operation, assuming the use of electronic buffering. The use of electronic buffers by optCA is considered first, since unlike optical delay lines, the duration which a packet can be stored in an electronic buffer is not bounded. This allows the use of protocols which could not be applied if optical buffering was used.

Another property of the sCA/B and sCA/R protocols, as well as many of the protocols previously developed for FT-TR WDM networks, is that every station must receive and process N mini-slots during every time slot. In this chapter, we consider an alternative whereby *only optCA* needs to receive and process N mini-slots. CA is responsible for destination conflict resolution in all CA-STAR networks. A by-product of the conflict resolution process is that CA would know which the packets that stations would be able to receive (i.e. those which would not be lost due to a destination conflict). The protocol considered in this chapter tries to take advantage of this by allowing optCA to inform each station of which packet it should receive by transmitting the channel index containing that packet in a mini-slot allocated for this purpose. This requires another N mini-slots, thus there are $2N$ mini-slots per control slot. The advantage is that each station now only has to receive and process one mini-slot per time slot—even if the networks size increases.

Section 5.1 describes the optCA central arbiter and the resulting network architecture named optCA-STAR. Section 5.2 is devoted to the development of a protocol for optCA-STAR networks. The protocol will be named optCA-MRS*. Following the convention introduced in Chapter 1, an optCA-STAR network operating according to the optCA-MRS* protocol will be called an optCA-MRS* network[1]. The performance of optCA-MRS* networks is analysed in section 5.3. Section 5.4 summarises our findings.

5.1 The Architecture of optCA-STAR Networks

The logical architecture of a CA-STAR network based on the optCA Central Arbiter (optCA-STAR network) is identical to that of the sCA-STAR network (see section 4.1 on page 70), except that the optCA central arbiter is used in

[1]The protocol is called optCA-MRS* because it is based on the MRS* algorithm, to be explained section 5.2

place of the sCA central arbiter.

5.1.1 The Structure of the Channels

Stations and optCA are synchronised, and channels are time slotted. The duration of a slot equals the transmission time of one (fixed length) packet, plus the tuning period [CHEN90], [HUMB93], [CHEN91], [CHLA91], [CHIP93], [PAPA92], [CHEN92].

Slots on channels λ_1, λ_2 , ..., λ_{2N} are called *data slots*. Each data slot can carry one data packet, Fig. 5.1.

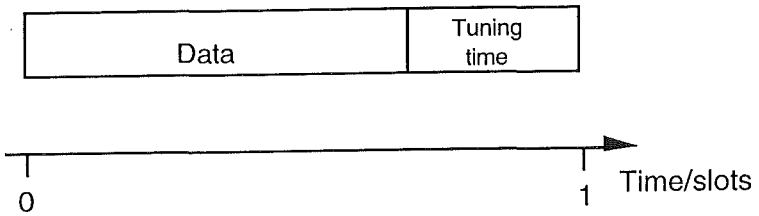


Figure 5.1: Format of a data slot.

λ_c is the common control channel. Slots on λ_c are called *control slots*. Each control slot is subdivided into $2N$ mini-slots, Fig. 5.2. Mini-slot i (i =1, ..., N) can carry the address of one station. Mini-slot j (j =N+1, ..., $2N$) can carry the index of a data channel[2].

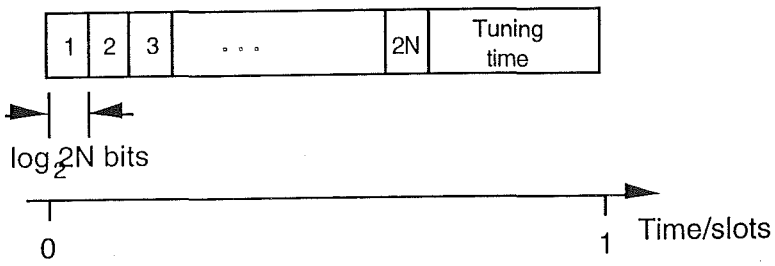


Figure 5.2: Format of a control slot.

Each station accesses its incoming and outgoing fibers through a network interface.

[2]The use of data and control slots, and other hardware elements, is protocol dependent. Therefore the specific uses of the data and control slots, the use of the FT_i and FT_c of stations and optCA, etc. will be specified in a given protocol specification.

5.1.2 Network Interface of Ordinary Stations

The network interface of ordinary optCA-STAR stations is identical to the network interface of stations in an sCA-STAR network. Accordingly, the transmission module of the network interface of station S_i operates two fixed tuned transmitters, see Fig. 5.4. One transmitter (FT_i) is for data transmission on λ_i. S_i (i =1, ..., N) uses λ_i as its own channel, dedicated for its data transmission. The other transmitter (FT_c) is tuned to the common control channel, λ_c.

Prior to transmission, packets are stored in a memory area of S_i called its transmit buffer. Ordinary stations need only a tiny transmit buffer, with a capacity for storing up to three packets[3].

The receiver module of S_i has one fixed tuned receiver (FR_c) for receiving from the control channel (λ_c), and one tuneable receiver (TR) for receiving data packets from any of the data channels.

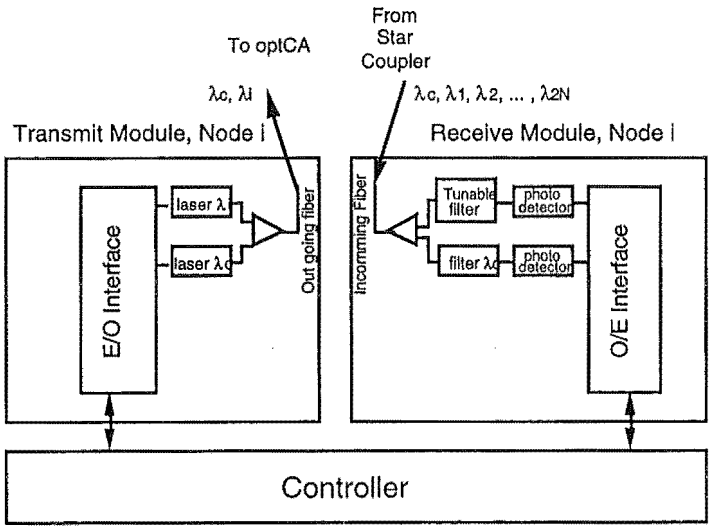


Figure 5.3: Block diagram of the network interface of stations in optCA-STAR networks.

[3]The use of hardware elements of optCA-STAR networks, including the buffer memory of ordinary stations, is protocol dependent. Therefore the uses of the buffer memory of ordinary stations and of optCA will be specified when a given protocol is described.

5.1.3 The optCA Station

As in sCA-STAR networks, the optCA central arbiter station is responsible for rescuing (buffering) packets that would otherwise be lost due to destination conflicts. optCA then re-schedules their arrival times so that they reach their destinations when their destinations are free to receive them.

Construction of optCA

optCA is made of N buffer modules Q_i (i=1, ..., N), plus one fixed tuned transmitter and receiver for accessing the common control channel, see Fig. 5.4. Each Q_i is made of one receiver R_i, one memory module M_i, and one fixed tuned transmitter T_i. The structure of a buffer module is shown in Fig. 5.5.

Signals from S_i arrive at the input I_i of optCA. They carry data on λ_i and control information on λ_c. Signals on λ_c from all inputs are coupled to the input of the control receiver. Signals on λ_i (of I_i) enters Q_i (i.e. the ith buffer module). The input into Q_i is split using a directional coupler. $1/(N+1)$ -th of the input power always enters R_i. R_i is for rescuing (receiving) packets arriving on λ_i which would otherwise be lost. Packets received by R_i are stored in M_i. These packets can be transmitted from M_i on data channel λ_{N+i}, using T_i.

The remaining $N/(N+1)$ -th fraction of the input power to Q_i is combined with the output from T_i. The combined signal then enters the ith port of the star coupler. The signal contains data packets on λ_i and λ_{N+i}.

The Difference Between the Design of optCA and sCA

With optCA, packets received by FR_i are stored directly into memory module M_i, from which they would be transmitted to their destination stations when appropriate. Unlike sCA, there could be at most one write (receive) and one read (transmit) access per memory module per time slot, regardless of network size. This means that the maximum memory access speed required of an optCA buffer equals that of ordinary stations. Consequently the buffer operations of optCA would not develop into the network's electronic bottleneck, even if N is increased. It should be noted that the use of multiple buffers per network interface is not an uncommon requirement. For example, some WDM networks assumes that every station is equipped with N-1 transmit buffers, one buffer for storing packets destined for a specific destination, e.g. [CHIP93].

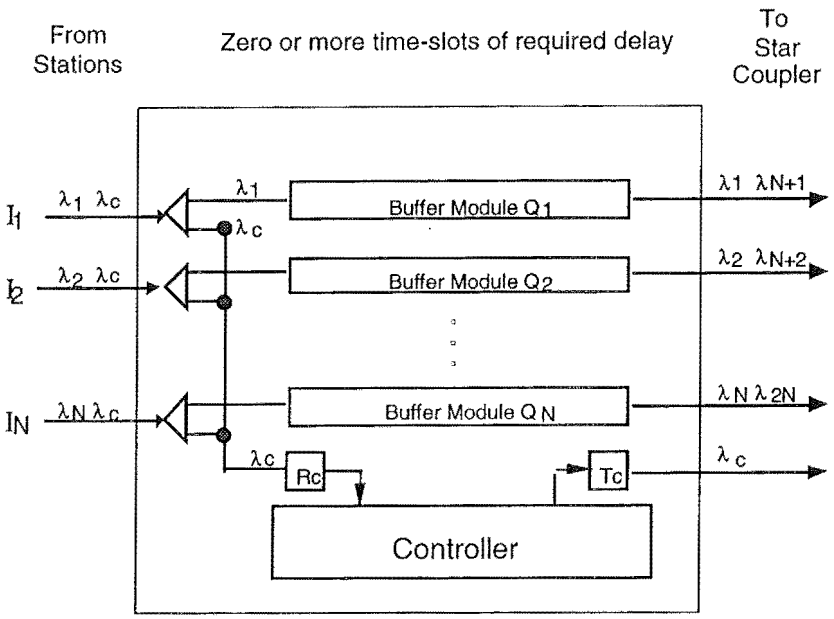


Figure 5.4: The optCA Conflict Arbiter station configuration for an optCA-STAR network with N stations.

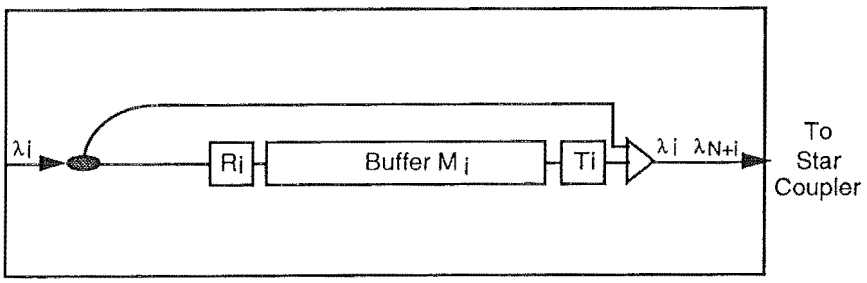


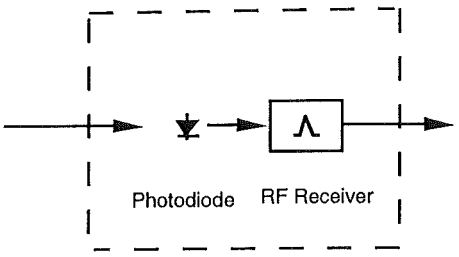
Figure 5.5: Block diagram of buffer module Q_i at optCA.

The busing structure of optCA is also drastically simpler than that of sCA. Only one read and write address/data bus is needed per buffer. Moreover, a time multiplexed address/data bus can be used, thereby further reducing bus count, if the addressing cycle could be completed during the tuning period of a time slot.

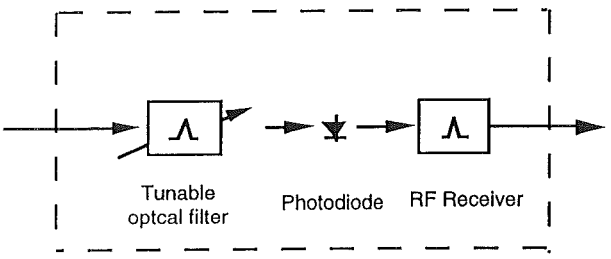
There are no data paths nor control lines between buffers in optCA. This separation between buffers appeals as optCA could be highly modularised. Unlike sCA, the optCA can be easily upgraded to serve a larger number of stations by adding one "buffer module" per additional station. optCA should also allow simpler fault diagnosis and graceful degradation. A fault with one bus, receiver, memory module, or transmitter, would only disrupt the

operations of one ordinary station (the one corresponding to the optCA buffer module with the faulty component), allowing all other stations to operate unaffected.

At the inputs to optCA, data channels are space division multiplexed (on separate fibers) instead of being wavelength division multiplexed. Resolving conflicts at optCA thus requires simpler hardware than resolving conflicts using multiple tuneable-filters/delay-lines/receivers at all ordinary stations, where incoming channels are wavelength division multiplexed. For instance, R_i is simpler than receivers at ordinary stations: an optical filter is not required, see Fig. 5.6.



(a) Reciever of optCA



(b) Receiver of ordinary stations

Figure 5.6: Block diagrams of receivers used by optCA and those used by ordinary stations in a WDM network.

Among the advantages of optCA, it is important to mention that the technologies needed for its construction are identical to that required for the buffers of ordinary CA-STAR stations. Any technological advancements (in the access time of semiconductor memories, buses, transmitters or receivers) that improve the data rate of ordinary stations should therefore also enable optCA to match the improved rate (of ordinary stations).

5.2 The optCA-MRS* Protocol

The general principle adopted by the optCA-MRS* protocol for optCA-STAR networks is similar to the principle followed by the protocols for sCA-STAR networks.

A ready station firstly announces its intention to transmit a packet on the control channel, then transmits the packet on its data channel after one time-slot. Destination conflicts occur if two or more packets are destined for the same destination during the same time slot. optCA detects destination conflicts, rescuing (buffering) packets that would otherwise be lost. optCA then schedules them for transmission to their to their destinations so that they arrive when their destinations are free to receive them. Ordinary stations receive just one mini-slot from the control during every time-slot, to determine which data channel to receive from during the next slot.

5.2.1 Structure of the optCA-MRS∗ Protocol

Ordinary stations of an optCA-STAR network are freed from the burden of destination conflict resolution, so their MAC protocol could be simplified. Thus, the optCA-MRS∗ protocol is defined by

1. the MAC protocol for ordinary stations, and
2. the MAC protocol of optCA.

The notation introduced in section 4.2.1 on page 77 for referring to time/data/control slots pertains.

5.2.2 MAC Protocol for Ordinary Stations

We assume that at ordinary stations, up to one new packet can be generated and transferred to the MAC layer per time slot, and that transfers are initiated at the beginning of a time-slot.

Define a packet to be *new* during t, if it was generated (transferred to the station's transmit buffer) during $t-1$. A packet in the transmission buffer during $t+1$ is said to be *waiting* if the station signalled (on the control channel) its intention to transmit it during t. A packet in the transmission buffer during $t+2$ is said to be *signalled* if the station signalled (on the control channel) its intention to transmit it during t.

Let S_i (i=1, 2, ..., N) maintain two variables. Both record the channel index received from the $N+i$th mini-slot (both of them are needed as will be explained below) :

1. Mini-slot value, h. h is an integer representing the channel index received from the $N+i$th mini-slot. $0 \geq h \geq log_2(2N)$.

2. Planned reception index, I. I=j during t if the station will receive the packet on channel λ_j during $t+1$.

Procedure Station Transmission (executed by S_i (i=1, 2, ..., N) during every time slot)

CoBegin
  if (S_i has a *new* packet) **then**
    transmit its destination address on the ith mini-slot of the current control slot.
    // That packet will be in the *waiting* state during the next time-slot, and in
    // the *signalled* state during the next-plus-one time-slot.
  if (S_i has a *signalled* packet) **then**
    transmit the packet on the current data slot of λ_i.
CoEnd ;

Procedure Station Reception (Executed by S_i (i=1, 2, ..., N) during every slot)

Begin
  CoBegin
    receive the index from the $N+i$th mini-slot and store it in h;
    if ($I \neq 0$) **then** receive the packet from λ_I ;
  CoEnd ;
$I = h$; tune receiver to λ_I during "tuning period" ;
End ;

Demonstration of the Packet Transmission and Reception Procedure of Ordinary Stations

The procedure used by ordinary stations for transmitting packets is demonstrated in Fig. 5.7, for a optCA-STAR network with 5 stations. S_2 signals its

intention to transmit a packet to S_4 by transmitting the destination address of the packet (i.e. 4) in the second mini-slot. That packet was generated during $t-1$, so S_2 signals its intention to transmit it during t. As shown, S_2 would transmit that packet during $t+2$. The state of the packet is *new* during t, *waiting* during $t+1$ and *signalled* during $t+2$.

The state transition diagram for packets is shown in Fig. 5.8. It is important to note that new packets change state deterministically from *new* to *waiting* to *signalled* during three consecutive time slots. If a packet is in the *signalled* state during a time slot, it will be transmitted during that slot[4]. This is why each new packet occupies the transmit buffer for exactly three time slots. As a result, a station needs only a small transmit buffer with a capacity for three packets.

S_2 transmits its signalling information on λ_c, and packets on λ_2.

Fig. 5.9 demonstrates the procedure followed by S_4 for receiving packets. During t S_4 listens on the incoming control channel. *S_4 only need to receive and decode the N+i-th mini-slot.* In this instance the channel index carried in the $N+i$th mini-slot equals 3. Accordingly, during the tuning period of the current slot S_4 tunes its receiver to λ_3, for receiving the packet that will arrive on λ_3 during $t+1$.

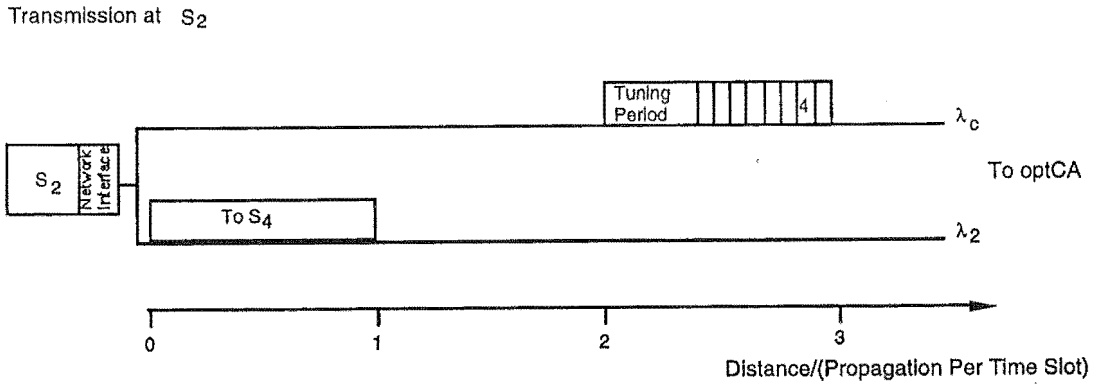


Figure 5.7: Packet Transmission Procedure, optCA-MRS* protocol.

[4]Thus a new packets in state *new* during the first time slot after its transfer to the MAC layer (its destination address is signalled on the control channel during that slot), in state *waiting* during the second time slot, and in state *signalled* during the third (during which it would be transmitted)

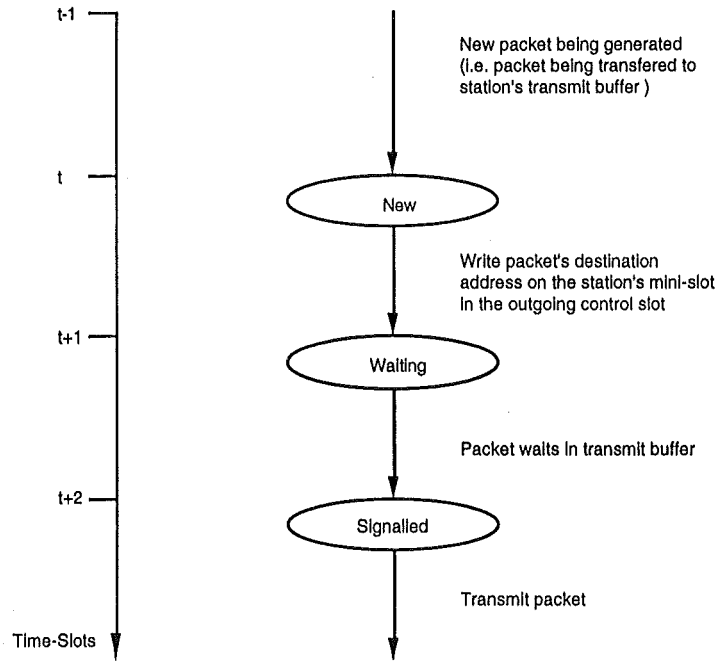


Figure 5.8: The deterministic state changes of packets, from *new* to *waiting*, to *signalled*.

5.2.3 The MAC Protocol of optCA

Using information it receives from the control channel, optCA determines which packets would be lost due to destination conflict(s), two time-slots prior to their arrival to optCA. optCA would then rescue (receive) such packets.

In addition, during every time slot optCA selects at most one packet from each of its buffer modules for conflict-free transmission during the next-plus-one time-slot. optCA informs ordinary stations in advance about which channel they should receive from, using mini-slots $N+1$, $N+2$, ..., $2N$ of each outgoing control slot.

Packets from S_i received by optCA are physically stored in buffer Q_i. There they await transmission to their destinations.

Under the optCA-MRS* protocol, packets stored in each physical buffer module are logically organised into N-1 FIFO queues. Denote the jth logical queue of buffer Q_i by $Q_{i,j}$. When a packet from S_i is rescued, it is enqueued in $Q_{i,j}$ of buffer module Q_i, if it is destined for S_j. Buffer Q_i's memory module is fully shared by its N-1 logical queues. Denote the length of $Q_{i,j}$ (including the packet being transmitted during the current slot, if any) by $L_{i,j}$, and denote

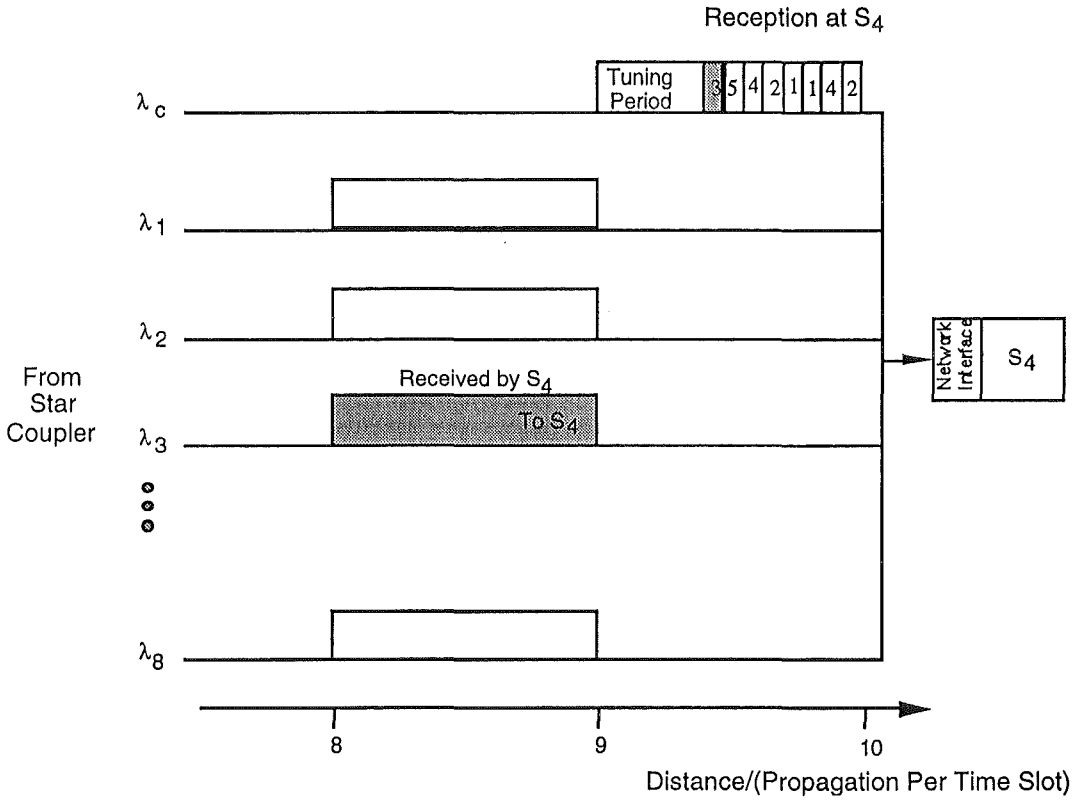


Figure 5.9: Packet Reception Procedure of ordinary stations, according to the optCA-MRS* protocol.

the total capacity of buffer module Q_i by B_i. Then, at any instant of time:

$$0 \leq L_{i,1} + L_{i,2} + ... + L_{i,N} \leq B_i, \tag{5.1}$$

and

$$0 \leq L_{i,j} \leq B_i, \quad for\ any\ j,\ j = 1, 2, ..., N,\ j \neq i. \tag{5.2}$$

We further assume that all buffer modules have the same capacity. Hence

$$B_i = B_j \quad for\ all\ 1 \geq i, j \geq N. \tag{5.3}$$

Let optCA maintain the following variables

1. Planned Reception Matrix, $P = [p_i\ ]_{N+1}$. At the start of t, $p_i = 1$ if during t, optCA should receive the packet transmitted by S_i; p_i =0 o.w..

2. Next Reception Matrix, $P^+ = [p_i^+]_{N\times 1}$. At the start of t, $p_i^+ = 1$ if during $t+1$ optCA should receive the packet from S_i ; p_i^+ =0 o.w..

3. Mini-slot Transmission Matrix, $M = [m_i]_{N+1}$. m_i, i=1, 2, ..., N, is the channel number which would be transmitted by optCA on the N+i-th mini-slot during the current time-slot.

4. Mini-slot Received Matrix, $H = [h_i]_{N\times 1}$. h_i i=1, 2, ..., N, is the address received from the ith mini-slot during the current time slot. h_i=0 if the ith mini-slot was empty.

5. Forward Address Matrix $F = [f_i]_{N\times 1}$. At the start of t, f_i=j indicates that optCA should transmit a packet from buffer Q_i to station S_j during t.

6. Next Forward Address Matrix $F^+ = [f_i^+]_{N\times 1}$. At the start of t, f_i=j indicates that optCA should transmit a packet from buffer Q_i to station S_j during $t+1$.

Procedure optCA Transmission (Executed during every slot)

CoBegin
  for $i = 1, 2, \ldots, N$ **do**
    transmit m_i on the N+ith mini-slot of the current control slot ;
  forall $i = 1, 2, \ldots, N$ **doparallel**
    if $(f_i > 0)$ **then**
      transmit packet from the head of Q_{i,f_i} ;
CoEnd

Procedure optCA Reception (Executed during every slot)

Begin
  CoBegin
    forall $i = 1, 2, \ldots, N$ **doparallel**
      if $(p_i \neq 0)$ **then** receive the packet from λ_i into Q_i ;
    receive mini-slots 1 to N and store their contents in H ;
  CoEnd ;
  $P = P^+$; $F = F^+$; Update M , P^+, and F^+ using MRS*(H) ;
End

The timing of optCA's control channel operations and the computation of MRS*(H) is specified in Fig. 5.10.

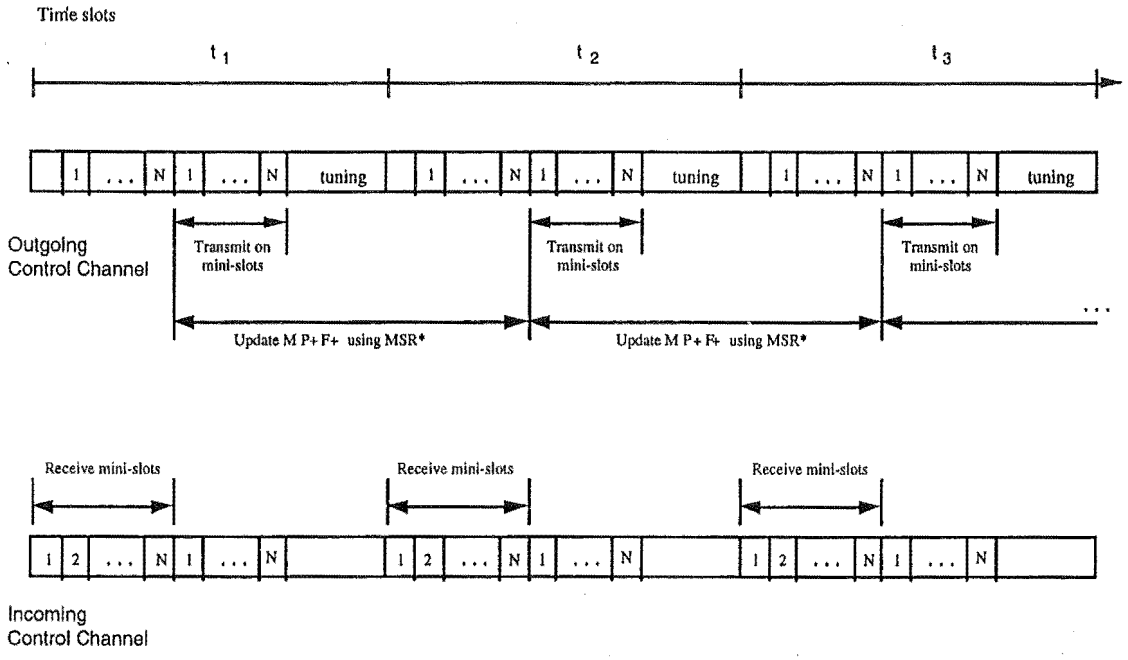


Figure 5.10: Timing of control channel operations and computation activities performed by optCA according to the optCA-MRS* protocol.

5.2.4 The MRS* Algorithm

The Maximum Remaining Sum (MRS) traffic assignment algorithm was proposed in [CHEN91] for use in "request-schedule-then-transmit" networks. In that application, MRS was executed by all ordinary stations during each time-slot to find a transmission schedule for the next time-slot [CHEN91].

Given the state of optCA's buffers, MRS can also be used to select packets for transmission (during the next-plus-one time slot) from optCA's buffers, subject to the following restrictions:

1. at most one packet may be chosen for transmission from each buffer; and
2. the chosen packets must have distinct destinations.

MRS is suboptimal because it cannot guarantee that it would select the maximum number of packets that could be transmitted, subject to the above constraints. However, it has been shown to be near optimal in benchmarks

[CHEN91], and require a smaller number of operations than the optimal algorithm[5].

In this section we introduce the MRS* algorithm which is an adaptation of the MRS algorithm proposed in [CHEN91]. During every time-slot, MRS* is invoked by optCA's reception procedure for three tasks. First, MRS* attempts to select packets for transmission (during the next-plus-one time slot) from optCA's buffers in a way which maximises the number of packets that can be received by stations, subject to the above-mentioned conflict-free restrictions. Secondly MRS* determines which incoming packets need to be rescued (received) by optCA. Thirdly, MRS* also updates M which is used to inform all ordinary stations of which channel they should listen to for data packet reception[6].

MRS* integrates these three tasks into one algorithm to avert replication of operations, thereby lowering the complexity of the optCA-MRS* protocol. MRS* was thus introduced for improved computational efficiency, not to improve the optimality of the MRS scheduling method. MRS* does take advantage of the fact that the set of packets received by ordinary stations during a time-slot can comprise of packets transmitted from ordinary stations, and "otherwise lost" packets transmitted by optCA.

MRS* records which packets optCA should transmit in F^+, and records which packets optCA should receive in P^+, and records which channel station S_i (i=1, 2, ..., N) should receive from in m_i.

MRS* maintains the following two static variables. They are static in the sense that they retain their values between invocations of MRS*.

1. Buffer Occupancy Matrix, $\beta = [\beta_{ij}]_{N \times N}$. During t, β_{ij} indicates the number of packets in buffer module Q_i during $t+2$ which would have S_j as their destination.

2. Destination Targeted Matrix, $D = [d_{ij}]_{N \times N}$. During t:

   $d_{ij} = 1$ if $b_{ij} > 0$, or if h_i=j , or both ;

   d_{ij}=0 o.w..

Let g_i be the ith column sum of D, and let w_i be the ith row sum of D. That is,

[5]Obviously, an algorithm is "optimal" if it would select the maximum number of packets that could be transmitted, subject to the above constraints

[6]optCA informs station S_i of the channel address by transmitting m_i on mini-slot $N+i$ of the control slot that arrives to S_i one time-slot ahead of the corresponding data packet, see Reception Procedure for ordinary stations above

$g_i = \sum_{k=1}^{k=N}(d_{ki})$

$w_i = \sum_{k=1}^{k=N}(d_{ik})$

Then the MRS* algorithm is defined as follows:

Procedure MRS*(H) ;

Begin

1. for i=1, 2, ..., N **do**
   if $h_i > 0$ **then** $p_i^+ = 1$; **else** $p_i^+ = 0$;
   F=[0,0, ...,0] ;
2. obtain D from β and H ;
3. obtain G=$[g_1, g_2, ..., g_N]$ and W=$[w_1, w_2, ..., w_N]$ from D ;
4. Find a smallest non-zero element in G and denote its index by q ;
   Find a smallest non-zero element(s) in W, denoting its index by p ;
   if $(W_p < G_q)$ **then** find all elements in W equal to the smallest, and select one of them at random, and assign its index to p ;
5. if $(W_p < G_q)$ **then** find new q such that $g_q = min_k\{g_k : d_{pk} \neq 0\}$;
   else find new p s.t. $w_p = randomly\ select\ one\ of \{min_k\{w_k : d_{kq} \neq 0\}\}$;
6. if $(\beta_{pq} > 0)$ **then** {
   (a) $f_p^+ = q$;
   (b) $m_q = N + p$;
   (c) W = W - $[d_{1q}, d_{2q}, ..., d_{Nq}]$;
   (d) **for** i=1 to N **do**
   if (($h_p \neq$ i) or $(i == q)$) **then** $g_i = g_i$ - $d_{p,i}$;
   (e) $d_{pq} = 0$;
   for i=1 to N **do**
   if (($h_p \neq$ i) or $(i == q)$) **then** { $d_{pi} = 0$; $d_{iq} = 0$; }
   (f) **if** ($(h_p \neq 0)$ and $(g_{h_p} \neq 0)$ **and** $(h_p \neq q)$) **then** $w_p = 1$;
   else $w_p = 0$;
   (g) $g_q = 0$; }
   else { // the packet transmitted by S_p to S_q would be received.
   (a) $p_p^+ = 0$;
   (b) $m_q = p$;

(c) $W = W$ - $[d_{1q}, d_{2q}, ..., d_{Nq}]$;

(d) $[d_{1q}, d_{2q}, ..., d_{Nq}] = [0, 0, ... , 0]$;

(e) $g_q = 0$; }

7. **if** $W \neq [0, 0, ... , 0]$ **then goto** 4. ;

for $i = 1, 2, ... , N$ **do**
{ **if** $(p_i^+ == 1)$ **then** $\beta_{i,h_i} = \beta_{i,h_i} + 1$;
if $(f_i^+ \neq 0)$ **then** $\beta_{i,f_i^+} = \beta_{i,f_i^+} - 1$; }

End

Comment :

- The random selection procedures in steps 4. and 5. were introduced so that stations are fairly treated.

5.2.5 Example of Network Operations According to optCA-MRS*

Consider a network with N=4 stations during time-slot t. Let the state of the network be as illustrated in Fig. 5.11. The packets in the buffer modules are represented by numbers of their destinations. The packets which will arrive or depart during t+1 are circled.

As shown, two packets will arrive to optCA during t+1. One will be rescued (received) into Q_1. The other will not be involved in a destination conflict, so it will bypass optCA and enter the star coupler. During t+1, previously rescued packets will be transmitted from Q_1, Q_2, and Q_3 for S_2, S_3, and S_4 respectively. Packets that will arrive to optCA during t+2 are also shown. Their destinations are known by optCA (recorded in H), from the mini-slots optCA received from the control channel during the initial part of t.

Fig. 5.12 shows how, during t, the MRS* algorithm use information stored in the Mini-slot Received matrix (H), to derive the Planned Reception matrix P^+, Planned Forwarding (optCA transmission) matrix F^+, and the Mini-slot transmission matrix M. The values of P^+ (F^+) at the end of t specifies the packets that optCA should transmit (receive) during t+2. At the end of t, the M matrix contains the values (channel indices) that optCA should transmit on mini-slots N+1 to $2N$, during t+1.

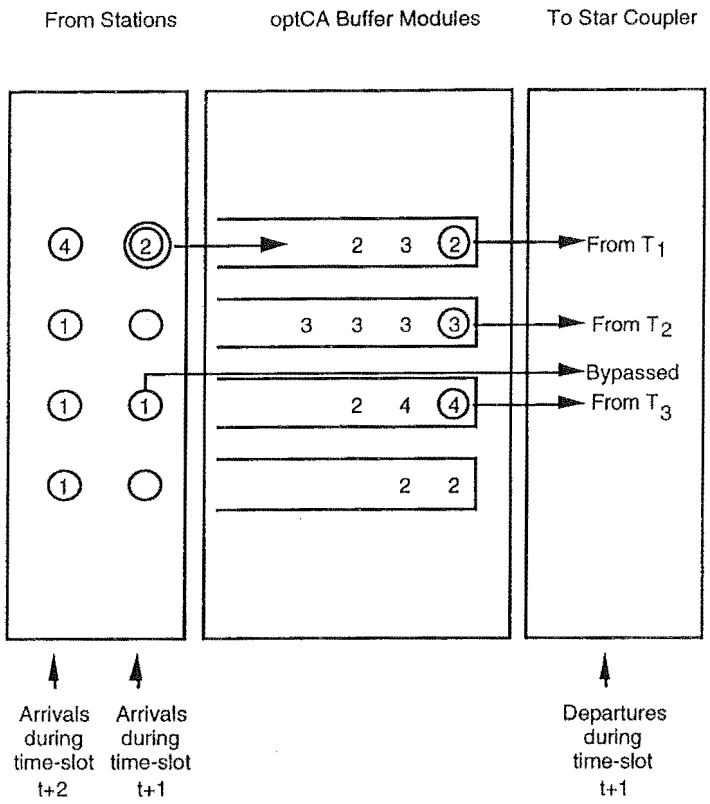


Figure 5.11: Example scenario at optCA during time-slot t. N=4.

The reception and the transmission procedure followed by optCA are demonstrated in Fig. 5.13 and Fig. 5.14, respectively. During t+1 optCA signals to stations the channels they should receive from, by transmitting M in mini-slots N+1 to $2N$. During t+2, optCA transmits packets designated by F^+ (during t), and rescues "otherwise-lost" packets designated by P^+ (during t).

5.3 Performance Analysis

5.3.1 The Model

We use the same model as that used in the analysis of sCA-STAR networks (see assumptions A1 to A6, section 4.2.7), and in [CHEN91], [PAPA92], [CHLA91], [CHEN90], [HUMB93], [CHEN92], and [CHIP93]. We consider a network of N stations. According to assumptions A1 to A6, each station generates new packets following an independent Bernoulli process, with probability p that

Initialisation

W=[4 1 1 1]
P^+=[1 1 1 1]
F^+=[0 0 0 0]
G=[3 3 2 2]
H=[3 2 3 2]

$$B=\begin{bmatrix}0 & (2+1-1) & (2-1) & 0\\ 0 & 0 & (4-1) & 0\\ 0 & 1 & 0 & (2-1)\\ 0 & 2 & 0 & 0\end{bmatrix} \quad D=\begin{bmatrix}0 & 1 & 1 & 1\\ 1 & 0 & 1 & 0\\ 1 & 1 & 0 & 1\\ 1 & 1 & 0 & 0\end{bmatrix}$$

Iteration 1

p=2 ; q=3 ;
new p = index of (randomly select one of min$\{h_1, h_2\}$)
= index of min$\{h_1$=3,h_2=2$\}$ = 2 ;
Since $B_{2,3} > 0$
f_p^+=3 ; => F^+=[0 3 0 0] ;
m_3=N+2 = 6 ; => M=[0 0 6 0] ;
$H = H$ - [1 1 0 0] = [2 2 3 2] ;
$G = G$ - [0 0 1 0] = [3 3 1 2] ;
h_2=1; => H=[2 1 3 2] ; g_3=0 ; => G=[3 3 0 2] ;

$$D=\begin{bmatrix}0 & 1 & \underline{0} & 1\\ \underline{1} & \underline{0} & \underline{0} & \underline{0}\\ 1 & 1 & \underline{0} & 1\\ 1 & 1 & \underline{0} & 0\end{bmatrix}$$

Goto 4 ;

Iteration 2

p=2 ; q=4 ; => new q =1 ;
Since B_{21}==0
p_2^+=0; => P^+=[1 0 1 1]; m_1=2 ; => M=[2 0 6 0] ;
H=H-[0 1 1 1] = [2 0 2 1] ;
g_1=0 ; => G=[0 3 0 2] ; Goto 4 ;

$$D=\begin{bmatrix}\underline{0} & 1 & 0 & 1\\ \underline{0} & \underline{0} & \underline{0} & \underline{0}\\ \underline{0} & 1 & 0 & 1\\ \underline{0} & 1 & 0 & 0\end{bmatrix}$$

Iteration 3

p=4 ; q=4 ; => new q=2 ;
Since $B_{4,2} > 0$
$f_3^+ f_p^+ = f_4^+$=2 ; => F^+=[0 3 0 2] ;
m_2=4+4=8 ; => M=[2 8 6 0] ;
H=H-[1 0 1 1] = [1 0 1 0] ;
G=G-[0 1 0 0] = [0 2 0 2] ;
h_4=0 ; => H=[1 0 1 0] ;
g_2=0 ; => G=[0 0 0 2] ; Goto 4 ;

$$D=\begin{bmatrix}0 & \underline{0} & 0 & 1\\ 0 & \underline{0} & 0 & 0\\ 0 & \underline{0} & 0 & 1\\ 0 & \underline{0} & \underline{0} & \underline{0}\end{bmatrix}$$

Iteration 4

p = randomly select one of $\{1, 3\}$ = 3 (say) ; q=4 ;
Since $B_{3,4} > 0$
f_3^+ = 4 ; => F^+=[0 3 4 2] ;
m_4=4+3=7 ; => M=[2 8 6 7] ;
H=H-[1 0 1 0] = [0 0 0 0] ;
G=G-[0 0 0 1] = [0 0 0 1];
g_4=0 => G=[0 0 0 0] ;

$$D=\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}$$

Since H=[0 0 0 0] Exit Loop
Updating B to account for planned receptions to and transmissions from buffer modules, gives

$$B=\begin{bmatrix}0 & 2 & 1 & 0\\ (0+1) & 0 & (3-1) & 0\\ 0 & 1 & 0 & (1-1)\\ 0 & (2-1) & 0 & 0\end{bmatrix}$$

Figure 5.12: Using the MRS* algorithm during t to plan packet receptions during t+2, mini-slot transmissions during t+1, and packet transmissions during t+2.

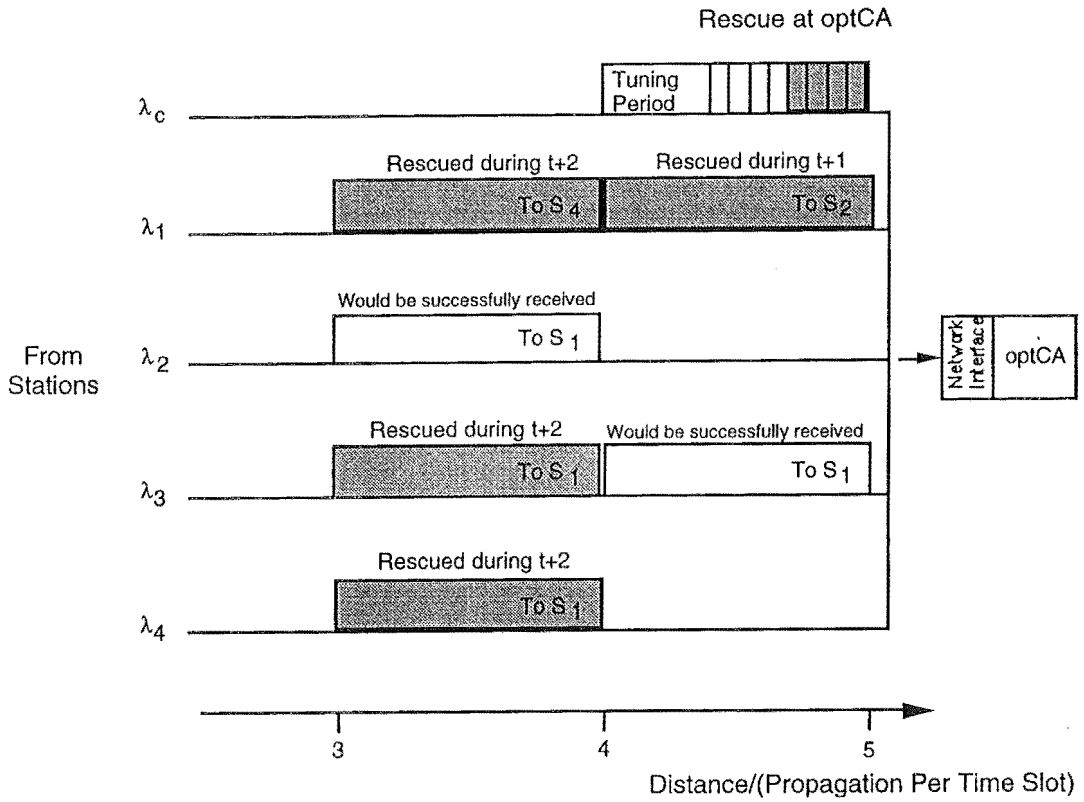


Figure 5.13: optCA packet reception procedure according to the optCA-MRS* protocol.

a new packet is generated at a given station during a time-slot. p will be referred to as the (normalised) load. The model assumes a uniform reference pattern, i.e. the destination of a packet is uniformly chosen over the N-1 other stations. We follow conventional notation by assuming that every station has a transmit buffer sized to hold B packets. In optCA-STAR buffer memories are used in optCA[7], giving each buffer module of optCA memory to store B packets. Let station S_i be a_i slots from optCA. The one way propagation delay is $2a$ slots when stations are equidistant from optCA.

5.3.2 Performance Measures

Performance characteristics of optCA-MRS* networks were obtained by simulating their steady-state behaviour, applying the methodology of quantitative stochastic simulation, discussed in section 4.2.7.

[7]Strictly, buffer memory for B-3 packets per station are centralised at optCA, assuming that memory for three packets is needed at each station. See section 5.2.2 for an explanation and example of why memory is needed for only three packets.

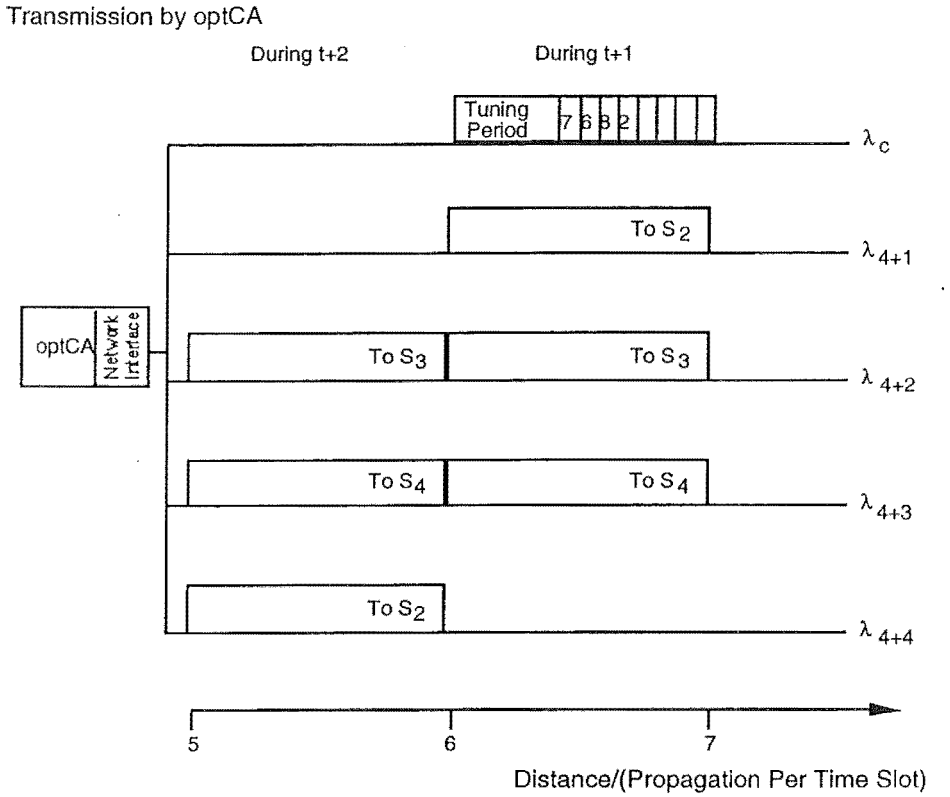


Figure 5.14: optCA packet transmission procedure according to the optCA-MRS* protocol.

The same performance measures of the sCA/B study will be considered. Namely:

- *normalised throughput* defined as the mean number of successful packet receptions per station per time slot; and
- *average packet delay* defined as the average of the time intervals from when a new packet is generated (i.e. transferred to the MAC layer) to when it is successfully received; and
- *mean excess delay of station* S_i where *excess delay* of a packet from S_i equals to its packet delay minus $a_i + a_d + 2$, ($a_i + a_d + 1$ equals the shortest possible delay of a packet measured in time-slots).

The SA-PTS methodology was developed for distributed execution of optCA-STAR simulations and for controlling the precision of the performance estimates during runtime [PAWL92], [YAUP93], [PAWL94] (discussed briefly in section 4.2.7, and described in detail in a separate technical report [YAU96a]).

5.3.3 Results

First consider optCA-MRS* networks with N=10 stations, where all stations are a=5 slots from optCA.

Effect of central buffer size on throughput/delay characteristics Figure 5.15 shows the (normalised) throughput as a function of the offered load for B=10, 15, 20, 30, 40.

It can be observed that the throughput of optCA-MRS* networks almost equals the offered traffic except at the highest traffic level, p=1, where throughput reached approximately 95% when B=10, and approximately 98.5% when B=40. In Fig. 5.16 the average packet delay is plotted as a function of the offered load. Provided that the offered load is below 90%, the mean delay experienced by packets is close to the minimum of one source-to-destination propagation delay plus two slots (12 slots), for all values of B considered. At p=1 delay has increased and is bounded by $0.5B+2a+2$.

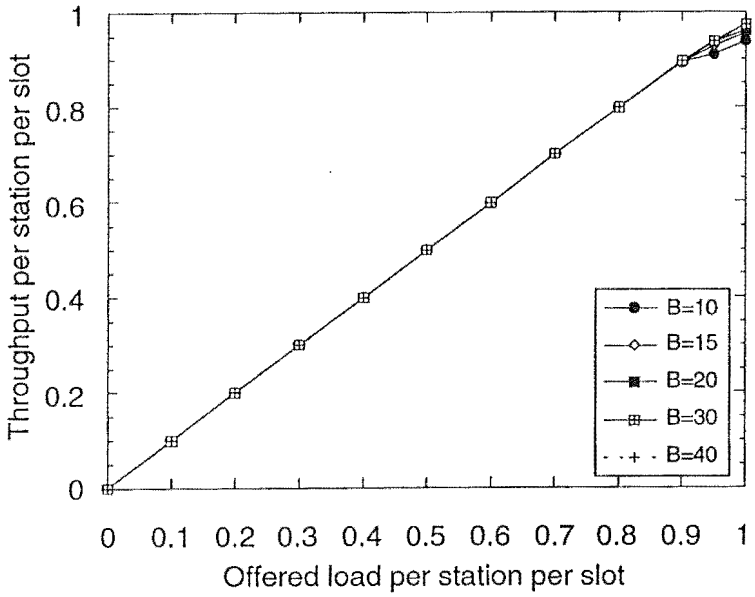


Figure 5.15: Throughput of optCA-MRS* versus load, for varying buffer sizes. N=10, a=5. Relative precision $\leq$5%.

Impact of increasing network size The effects of increasing the network size were explored by considering optCA-MRS* networks with N=3, 5, 10, 20, 40, and 100 stations. B=25 in all cases. Results are graphed in Fig. 5.17 and Fig. 5.18. One can see that increasing the number of stations in the optCA-MRS* network from 10 to 100 does not affect its efficiency. Increasing network size from 3 to 10 stations increases the average packet delay somewhat

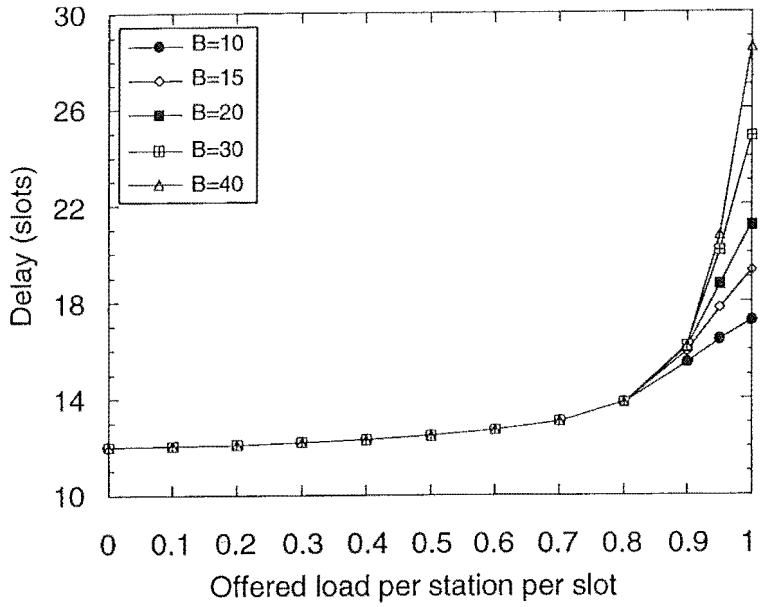


Figure 5.16: Average packet delay of optCA-MRS* versus load, for varying buffer sizes. N=10, a=5. Relative precision $\leq$5%.

at medium traffic, see Fig. 5.18, but further increases have little effect. These results demonstrate that optCA-MRS* remains a good solution as network size increases, *even when B is unchanged.*

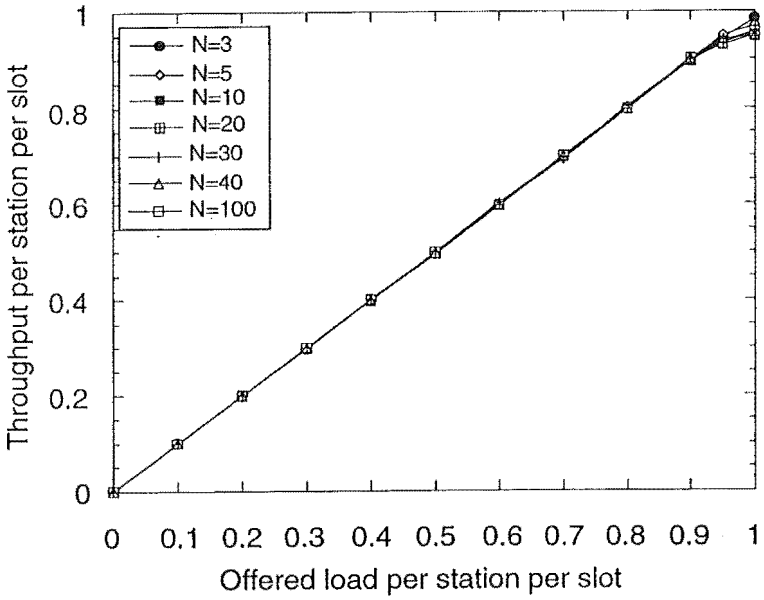


Figure 5.17: Throughput of optCA-MRS* versus load, for varying number of stations. B=25, a=5. Relative precision $\leq$5%.

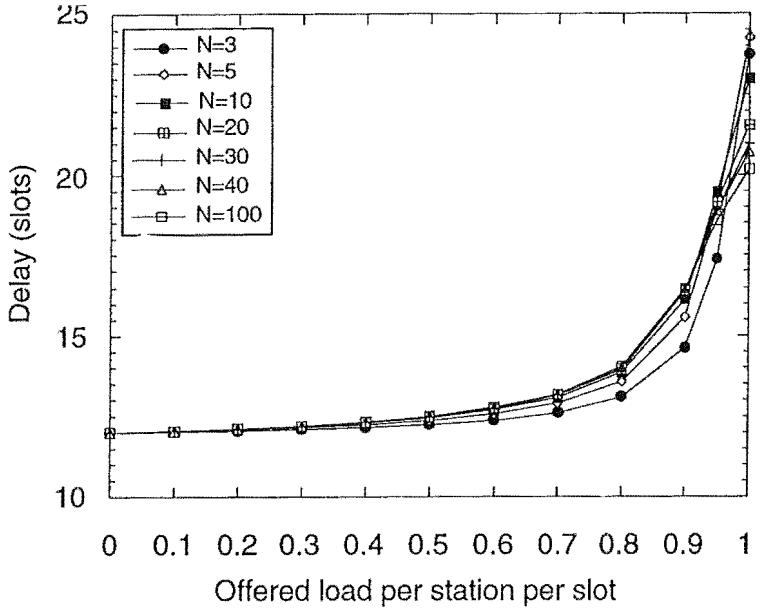


Figure 5.18: Mean packet delay of optCA-MRS* versus load, for varying number of stations. B=25, a=5. Relative precision $\leq$5%.

Heavy traffic behaviour Since the throughput of optCA-MRS* with B=10, 20, 30, 40, were almost indistinguishable from ideal (i.e. throughput almost equalled offered load) except at the highest possible traffic level, an investigation of throughput/delay when p=1 for a wider range of B is a natural next step. Results for an optCA-MRS* network with N=10 stations are reported in Table 5.1. The first column gives the buffer size (B) per station. The second and third gives the point estimates of the probability of packet loss followed by their final confidence intervals at the 0.95 level. The throughput estimated as 1-(estimate of the probability of packet loss) is shown in the fourth column. The mean excess delay estimates and their confidence intervals are presented in columns five and six.

An examination of Table 5.1 shows that the normalised throughput of the network increased from 82.6% when B is 2, to 93.6% when B is 10. Increasing B above B=10 yields diminishing returns.

Excess delay increases with increasing B as well. The positive relationship between B and average packet delay can be due to the fact that at maximum offered load, increasing B increases the number of packets that could be accepted into the network from stations' upper (LLC) layers. Yet the network's reception capability is saturated, so a larger number of admitted packets means a longer wait before being delivered, as expected from Little's result.

B	P(Packet loss)		Throughput	Mean Excess Delay	
2.000	0.1745	(0.1730, 0.1760)	0.8255	1.225	(1.214, 1.236)
4.000	0.1146	(0.1139, 0.1153)	0.8854	2.405	(2.384, 2.425)
6.000	0.08918	(0.08856, 0.08980)	0.9108	3.418	(3.390, 3.446)
8.000	0.07398	(0.07347, 0.07450)	0.9260	4.344	(4.303, 4.386)
10.00	0.06444	(0.06403, 0.06485)	0.9355	5.193	(5.155, 5.231)
12.00	0.05663	(0.05630, 0.05696)	0.9434	6.013	(5.956, 6.070)
14.00	0.05153	(0.05127, 0.05178)	0.9485	6.803	(6.759, 6.846)
16.00	0.04685	(0.04641, 0.04729)	0.9532	7.590	(7.517, 7.663)
18.00	0.04311	(0.04291, 0.04331)	0.9569	8.331	(8.263, 8.400)
20.00	0.04008	(0.03969, 0.04046)	0.9599	9.133	(9.051, 9.215)
30.00	0.03034	(0.03015, 0.03053)	0.9697	12.84	(12.76, 12.91)
40.00	0.02507	(0.02483, 0.02530)	0.9749	16.54	(16.39, 16.69)
50.00	0.02211	(0.02193, 0.02230)	0.9779	20.31	(20.16, 20.46)
60.00	0.02002	(0.01985, 0.02018)	0.9800	24.12	(23.88, 24.35)
80.00	0.01754	(0.01740, 0.01768)	0.9825	31.56	(31.31, 31.81)
100.0	0.01621	(0.01606, 0.01636)	0.9838	39.40	(39.02, 39.78)
200.0	0.01392	(0.01382, 0.01403)	0.9861	75.10	(74.40, 75.80)
300.0	0.01348	(0.01340, 0.01357)	0.9865	114.1	(113.3, 115.0)
400.0	0.01327	(0.01314, 0.01339)	0.9867	145.6	(144.3, 146.9)

Table 5.1: Performance of optCA-MSR* networks under maximum load, as a function of B. p=1, N=10, a=5.

The performance of optCA-MSR* networks as a function of p near maximum load is reported in Table 5.2. An examination of Table 5.2 shows that the probability of packet loss diminishes rapidly as the offered load is reduced from p=1. This suggests that optCA-MSR* can provide good throughput/delay/packet loss performance, provided that the offered load is not too high.

5.3.4 Computational Complexity Analysis

Define *network computational complexity* of a MAC protocol as the maximum number of scalar operations for MAC purposes that is performed in the network during one time-slot. Let the network computational complexity of protocol A be denoted by C_N(A).

Define *time computational complexity* as the maximum number of time steps needed for executing the MAC protocol by the most MAC computa-

p	P(Packet loss)		Throughput	Mean Excess Delay	
1.000	0.04008	(0.03969, 0.04046)	0.9599	21.13	(21.05, 21.215)
0.9900	0.03308	(0.03276, 0.03340)	0.9669	20.67	(20.61, 20.731)
0.9800	0.02665	(0.02639, 0.02691)	0.9733	20.22	(20.14, 20.292)
0.9700	0.02067	(0.02047, 0.02088)	0.9793	19.79	(19.74, 19.855)
0.9600	0.01525	(0.01512, 0.01537)	0.9848	19.27	(19.20, 19.338)
0.9500	0.01059	(0.01051, 0.01067)	0.9894	18.71	(18.66, 18.761)
0.9400	0.006905	(0.006852, 0.006957)	0.9931	18.19	(18.15, 18.221)
0.9300	0.004146	(0.004114, 0.004178)	0.9959	17.60	(17.55, 17.652)
0.9200	0.002259	(0.002238, 0.002279)	0.9977	17.09	(17.04, 17.130)
0.9100	0.001138	(0.001129, 0.001146)	0.9989	16.57	(16.52, 16.609)

Table 5.2: Performance optCA-MSR* networks as a function of p, near maximum load. B=20, N=10, a=5.

tional intensive station in the network, during one time slot. Denote the time computational complexity of protocol A by C_T(A).

As in [CHEN91], [CHEN92], [CHEN94], we assumed that each scalar operation can be done in one time step. Also following [CHEN91], [CHEN92], [CHEN94], an assignment, comparison, addition or subtraction operation will be considered as a scalar operation.

C_N(A) can, for example, be used to compare the aggregate electronic processing overhead. In the networks considered, the MAC algorithm has to be executed within under one, or at most several time-slots. If we are concerned that the time required for executing protocol A may create a bottleneck, then C_T(A) may be a more relevant complexity index.

The time and the network computational complexities of optCA-MSR* networks (en route conflict resolution), sCA/B networks (en route conflict resolution), CF-WDMA (request-schedule-then-transmit) networks, DT-WDMA (detect-and-retransmit-if-lost) networks, and the DAS (request-schedule-then-transmit) and HTDM (hybrid request-schedule-then-transmit/fixed transmission schedule) networks are derived in Appendix F.2, F.1, F.3, F.4, F.5, and F.6 respectively. The results are collected in Table 5.3.

Network	Computational Complexity	
	C_T	C_N
Request-schedule-then-transmit (CF-WDMA)	$20N^2 + 3N$	$20N^3 + 3N^2$
Detect-and-retransmit (DT-WDMA)	$22N + 15$	$22N^2 + 15N$
Request-schedule-then-transmit (DAS)	$\sum_{i=1}^{N}\sum_{j=1}^{N}(N-i+2)(N-j+2)(N+i-1)$	$N\sum_{i=1}^{N}\sum_{j=1}^{N}(N-i+2)(N-j+2)(N+i-1)$
Request-schedule-then-transmit (HTDM)	$\frac{M}{N+M}\sum_{j=1}^{N}\sum_{k=1}^{N}(N-j+2)(N-k+2)(N+j-1)$	$\frac{NM}{N+M}\sum_{j=1}^{N}\sum_{k=1}^{N}(N-j+2)(N-k+2)(N+j-1)$
sCA-Star (sCA/R)	$8N^2 + 18N + 2$	$12N^2 + 21N + 2$
optCA-STAR (optCA-MRS*)	$23N^2 + 25N$	$23N^2 + 29N$

Table 5.3: Computational complexities of MAC protocols of various WDM Star networks.

One can see that optCA-MRS* has the same or lower order of time complexity as that of networks based on the "request-schedule-then-transmit" method. However, optCA-MRS* has a higher order of time complexity when compared with the "detect-and-retransmit" network, and the sCA/B sCA-STAR network.

5.4 Chapter Conclusions

The optCA central arbiter was introduced as an alternative to sCA for detecting destination conflicts, and buffering all packets which would otherwise be lost[8], re-scheduling their arrival times so that they reach their destinations when their destinations are free to receive them. optCA has simpler busing structure and physical memory organisation than the sCA. The maximum memory access speed required for the optCA buffer modules equals that of the transmit buffer of ordinary stations. Consequently the buffer operations of optCA would not develop into the network's electronic bottleneck, even if

[8]When more than one packet simutaneously arrive for the same destination, the destination can receive only one of them. The other packets can therefore be considered as "otherwise lost" packets which needs to be rescued by CA.

the number of stations is increased. Unlike an electronic switch or a station in a multihop WDM network, optCA performs neither external routing, space switching, nor internal queuing. optCA is modularised, with one buffer module serving each station. There are no data paths nor control lines between buffer modules in optCA. optCA can be upgraded to serve a larger number of stations by adding one "buffer module" per additional station. optCA should also allow simpler fault diagnosis and graceful degradation. A fault with one bus, receiver, memory module, or transmitter, would only disrupt the operations of one ordinary station (the one corresponding to the optCA buffer module with the faulty component), allowing all other stations to operate unaffected. The technologies needed for the construction of optCA are identical to that required for the network interface of ordinary CA-STAR stations. Any technological advancement (in the access time of semiconductor memories, bandwidth of buses, or the rate of transmitters or receivers) that improve the data rate of ordinary stations should therefore also enable optCA to match the improved rate.

The optCA-MRS$*$ protocol was proposed for optCA based CA-STAR networks. The preceding analysis showed that optCA-MRS* networks offer very good performance. When $B = 10$, the throughput provided by optCA-MRS* can be very close to 100%, and the mean packet delay was very close to the minimum, provided the offered load was below 90%.

The optCA-MRS* protocol has the same or lower order of time computational complexity ($O(N^2)$) as that of the CF-WDMA, HTDM, and DAS protocols which were developed for "request-schedule-then-transmit" networks. On the other hand, the time computational complexity of optCA-MRS$*$ is higher than that of the protocol for "detect-and-retransmit" networks. The MAC protocol of the later have a time computational complexity of just order $O(N)$. The performance congruency of optCA-MRS$*$ with sCA/B networks suggests that the simpler optCA can achieve a very comparable level of performance with that of sCA.

Chapter 6

Conflict-free Traffic Assignment using Forward Planning

6.1 Introduction

A drawback of optCA-MRS* is the high computational complexity of the conflict-free scheduling algorithm used by the MAC protocol of optCA. If the buffer modules of optCA transmitted the "otherwise lost" packets in a random order, then the problem of destination conflicts amongst packets (transmitted by optCA) would arise whenever two or more modules transmits packets to the same destination during the same time slot. The optCA station uses the MRS* algorithm to ensure that packets it "rescued" are transmitted in a destination conflict-free way.

However, the choice of the MRS algorithm is not an immediate consequence of the use of optCA. MRS is only one possible solution to the problem of scheduling the transmission of packets to their destinations in the situation where packets originating at station S_i are stored in input buffer Q_i, and each destination station could receive only one packet during one time slot. This conflict-free scheduling problem also arises in "request-schedule-then-transmit" WDM star networks [CHEN91], [CHIP93], [CHEN92], [CHEN94], [FOOR95], [YAU96b], and therefore any of the previously proposed algorithms can perform the scheduling function for optCA. However, an algorithm *able* to perform this function differs greatly from one that is *well suited* for the task. The aim is therefore to develop an algorithm in light of the special environment in which the scheduler would be implemented. The CA-STAR setting for the scheduling problem differs from request-schedule-then-transmit

networks in terms of the *location(s)* where scheduling is performed, and the *time* when it is performed in relation to the packet exchange process.

The CA-STAR and request-schedule-then-transmit WDM star networks can be logically modelled by an interconnection system made of a non-blocking switch with N inlets and input queues, each with a buffering capacity of B packets, and N outlets, see Fig. 6.1. An inlet represents a buffer module in the case of optCA, or a source station in the case of a "request-schedule-then-transmit" WDM network. Packets are queued at the inlet, waiting for transmission to their target outlets (destination stations). Let us refer to a CA-STAR/request-schedule-then-transmit WDM star network with N inlets[1] and outlets, where each inlet queue has a capacity of B packets, as a $N \times B$ interconnection system (IS).

This chapter is concerned with the design of a traffic assignment (scheduling) algorithm for selecting packets queued at the inlets for transmission to their intended outlets in a conflict-free way, thereby improving throughput of the IS. Major concerns in the CA-STAR context include its throughput-delay characteristics, computational complexity, and the physical and logical buffer organisation that is required for its implementation.

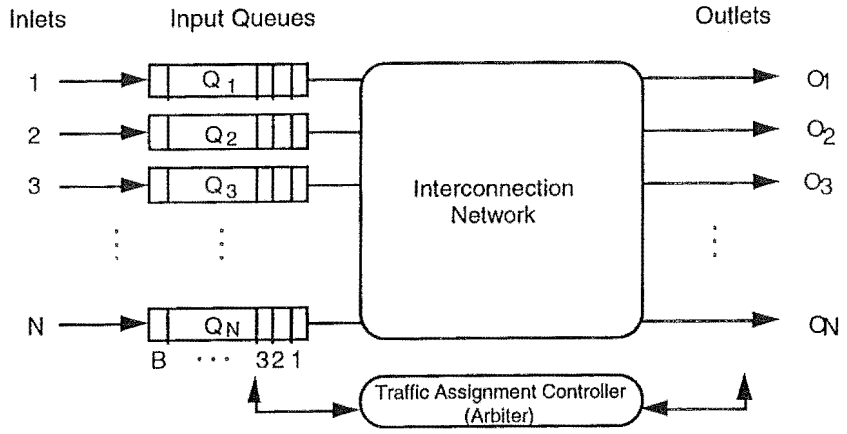


Figure 6.1: Model of an $N \times B$ Interconnection System (IS).

Several algorithms have already been introduced for achieving conflict-free traffic assignment in such systems. All these algorithms are for execution during every time-slot. During a given time-slot, the algorithm examines packets in the inlet queues, attempting to maximise the number of pack-

[1]In previous chapters, N denoted the number of stations. An inlet represents either a station or a buffer module, so the number of stations or buffer modules equals the number of inlets. Thus it is natural to use denote the number of inlets here by N

ets which may be transmitted conflict-free during the next time-slot. Of these, the SDR (System of Distinct Representatives) algorithm [CHEN91], [INUK79], offers the best throughput and delay performance. SDR is *optimal* in that, given the $N \times B$ system state, it maximises the number of queued packets for transmission during a time-slot, subject to the conflict-free and ordering constraints. The high computational complexity of SDR has motivated the development of suboptimal but less complex algorithms. One of them, the MRS (Maximum Remaining Sum) algorithm [CHEN91], [CHEN92] was originally proposed for transmission scheduling in "request-schedule-then-transmit" lightwave networks, see section 2.3.2. MRS has also been adopted by the optCA-MRS* protocol (see the previous chapter). MRS uses a heuristic for selecting as many packets as possible for conflict free transmission during one time-slot. An interesting alternative is the K-HOL (K-Head-Of-Line) algorithm [CHEN94] which takes only the first K, $K \leq B$, packets at each inlet queue into account, when making a traffic assignment during a time-slot. Limiting the number of packets considered significantly lowers the complexity of K-HOL, although it also lowers the performance of the IS in the sense of throughput and delay. The RS (Random Selection) algorithm (also proposed for use in request-schedule-then-transmit networks [CHIP93]) uses a heuristic which selects packets randomly, checking each one with selections of previous iterations, accepting it for transmission only if there is no destination conflict.

The Forward Planning Conflict Free (FPCF) algorithm that is proposed in this chapter tries to achieve the same goals of high throughput and low computational complexity, using a different concept. FPCF plans the transmission of all (buffered) packets for up to B-1 slots into the future. In physically realisable switches, input buffer capacity is finite, so new packets are lost when the buffer is full. Knowing which packets are to be transmitted in the next B-1 time-slots allows FPCF to accept new packets into buffers only if they could be transmitted conflict-free in the foreseeable future. By selectively rejecting new packets, valuable buffer "real-estate" is retained so more new packets that could be transmitted conflict-free could be accommodated. Reduced rate of lost packets and improved throughput and delay performance is thus expected.

With FPCF, a new packet arriving during the current time-slot has its transmission scheduled for one of the future B-1 time-slots, or is rejected. Thus all buffered packets have had their transmissions already scheduled. Consequently, unlike previous solutions where the entire traffic matrix of buffered packets is considered during each slot when planning the next transmissions, in FPCF only new packets need to be processed during each time-slot. This allows the computational complexity of the algorithm to be drasti-

cally lowered.

Section 6.2 describes in detail the above mentioned traffic assignment problem. Section 6.3 is devoted to the definition of the FPCF algorithm. Section 6.4 introduces the model assumed in the performance analysis of the proposed algorithm, followed by the actual results. The SDR algorithm has the highest performance of the previous algorithms. We therefore compare the performance FPCF with SDR, and then compare the computational complexity of FPCF with SDR, and with the various approximate algorithms. The final section focuses on some implications for CA-STAR networks.

6.2 Problem Specification

This study considers the generic traffic assignment problem in $N \times B$ Interconnection Systems, see Fig. 6.1. We assume time-slotted systems. Each slot equals the (fixed) service time of one customer, called a packet. Packets may arrive at inlets at the start of time slots only.

If a packet is targeted for outlet O_j, then O_j is said to be its destination. Packets queued at the inlets are transferred to their outlets via a logical $N \times N$ switch. The specific set of packets transferred during one time-slot is determined by a *traffic assignment algorithm.* The following traffic assignment rules have to be observed in practical applications :

R1 at most one packet from each inlet may be chosen for transfer during one time slot,

R2 at most one packet may be destined to any given outlet during one time-slot (to avoid a destination conflict); and

R3 packets queued at inlet queue[2] Q_i destined for outlet O_j must be transferred in their relative order of arrival to Q_i (thus, the original order of packets has to be always maintained).

Let the system state be represented by the System State Matrix $S = [s_{ij}]_{N \times B}$; where $s_{ij}=k$ $(1 \geq k \geq N)$ if a packet stored in the jth location of the ith buffer is destined for outlet k, and $s_{ij}= 0$ if this location is empty. Since

[2]In previous chapters, Q_i denoted memory of the i-th buffer module. As the queue of inlet i represents either the i-th buffer module (case of optCA-STAR) or the i-th station (case of a request-schedule-then-transmit network), a natural choice of the symbol for the queue of inlet i is Q_i.

each row of S represents B packet-size *memory locations* of one buffer, there can be at most one "write" (receive) and one "read" (transmit) operation per row during a time slot, and a read and write on the same row must be on distinct locations.

6.3 The Forward Planning Conflict Free (FPCF) Algorithm

Following FPCF, the packets transmitted during a given time slot are determined by the E-th column of S, i.e. by [$s_{1,E}$, $s_{2,E}$, ..., $s_{N,E}$], $1 \geq E \geq B$. Namely, if $s_{i,E} \neq 0$, then the packet in the E-th location of buffer Q_i is transmitted. The column index E rotates after each time-slot. Thus, with FPCF, the *traffic assignment* problem becomes the problem of *packet placement.* When a packet arrives to the i-th inlet, it is placed in the first-to-be-served empty position in Q_i, whilst meeting the constraint that packets in the same location in buffers Q_1, Q_2, ..., and Q_N have to have distinct destinations. Let the algorithm maintain the following variables[3] :

1. System State Matrix $S=[s_{i,j}]_{N \times B}$; defined above.
2. Destination Allocation Matrix, $\Delta = [\Delta_{ij}]_{N \times B}$. $\Delta_{ij}= 1$ if one of s_{1j}, s_{2j}, ..., s_{Nj} equals i. $\Delta_{ij}= 0$ o.w.
3. FaVoured Station (V) Counter; $1 \geq V \geq N$. Initialised to 1 during network startup.
4. Enabled Column Counter, E; $1 \geq E \geq B$. Initialised to B during system startup, and decremented by one after each time slot; if $E==0$ then E is reset to B.

Also, let $H = [h_i]_{N \times 1}$ be an array of destination addresses of (up to) N packets arriving during the current time slot. Thus, $h_i = j$ if a packet destined for outlet O_j arrives at inlet i. Such a new packet is referred to as π_i . If no packet arrive at inlet i then $h_i = 0$.

The FPCF algorithm can be defined by the following pseudo code:

[3]Normal scope rules apply. Thus variables declared for the definition of the FPCF algorithm are visible only within its definition. A variable with a given name is as declared in the algorithm specification, even if the same name have been used to identify another variable in another protocol/algorithm/procedure.

procedure FPCF(input H) // Plan transmissions (packet placement)
BEGIN
$V = V + 1$; **if** $(V > N)$ **then** $V = 1$;
$E = E$ -1; if $(E == 0)$ then $E = B$;
for each inlet i, $i = V, V+1, ..., N, 1, 2, ..., V-1$ **do**
if $h_i \neq 0$ **then** // new packet arrives to inlet i
{ **find** the first j s.t. ($s_{ij} == 0$ **AND** $\Delta_{h_i,j} == 0$, searching in the order given by $j = E-1, ..., 1, B, B-1, ..., E+1$;
if found, **then** { store packet π_i in jth location of buffer Q_i; $s_{ij} = h_i$; $\Delta_{h_i,j} = 1$; } **else** discard packet π_i ;
}
endfor
$[s_{1E}, s_{2E}, ..., s_{NE}] = [0, 0, ..., 0]$; $[\Delta_{1E}, \Delta_{2E}, ..., \Delta_{N,E}] = [0, 0, ..., 0]$;
END;

The role of counter V is to rotate the order in which inlets are processed during each slot. The sooner an inlet is processed, the greater the probability that its new packet (if any) could be buffered. Rotating the processing order ensures that inlets are fairly treated. Ordinarily, during the time-slot when FPCF is executed, the new packets are assumed to be held in temporary storage at their inlets. But in CA-STAR networks, FPCF is executed (by optCA only) *prior* to the packets' arrival to optCA. Thus packets can be stored directly in their planned physical memory locations: temporary storage is unnecessary. It is important to note that FPCF preserves the order of packets arriving at a given inlet and destined for the same outlet. In contrast to previous algorithms, no special computation steps, nor the logical organisation of buffered packets into FIFO queues or lists, or any other logical structure, is necessary.

Realisable systems have finite input buffer capacity, so new packets are lost when no suitable buffer space can be found at the time of their arrivals. With FPCF, the transmission of all (buffered) packets is planned for up to B -1 time slots into the future. Using knowledge of future transmissions (recorded in S), FPCF deduces whether a new packet could be transmitted conflict-free in the foreseeable future (using Δ and S) when deciding if the packet should be buffered. By selectively accepting new packets into the buffers (instead of discarding new arrivals indiscriminately when the buffers are full) valuable buffer "real-estate" is retained so more new packets that could be transmitted conflict-free could be accommodated. Furthermore, only packets arriving during the current time-slot need to be scheduled, since all packets left in buffers from the previous time-slot have been already scheduled for conflict-free transmissions. An improvement in throughput (reduction in probability

of packet loss) and a substantial reduction in computational complexity is thus expected.

Another nice feature of FPCF is that it requires extremely simple buffer organisation. Unlike other algorithms, no logical structures of packets in an inlet buffer (e.g.. no FIFO queues of packets, list, or multiple queues) need to be created or maintained. Also the (deterministic) transmission of up to N packets per time slot, pointed out by the pre-selected (Eth) column of S, permits somewhat simplified access modes.

6.4 Performance Analysis

This section evaluates the FPCF algorithm in terms of performance, worst case computational complexity, and buffer organisation complexity.

The SDR algorithm enjoys the best throughput performance of the previously introduced algorithms. The MRS, K-HOL, and RS algorithms have lower performance, but feature lower worst case computational complexity.

Therefore, the approach should be to compare the throughput and delay characteristics of FPCF with SDR. Then the worst case computational complexity and the buffer organisation complexity of FPCF will be compared with that of SDR, as well as the MRS, K-HOL, and RS algorithms.

6.4.1 The Traffic Model

All algorithms considered here are analysed assuming the same traffic model as in [CHEN90], [YAUP92], [CHEN91], [PAPA92], [CHLA91], [HUMB93], [CHEN92], [CHEN94], [YAU94], [CHIP93], and [YAU96]. Specifically, consider a $N \times B$ system. New packets arrive at inlets following N independent and identical Bernoulli process, with probability p of having a new packet arriving at a given inlet during a time slot, $0 \geq p \geq 1$. p will be referred to as the (normalised) offered load. A uniform reference pattern is assumed, i.e. destinations of packets are uniformly distributed over the set of N outlets.

The situation represented by this model differs from that at the optCA station of optCA-STAR networks (see section 5.3) in the following ways.

1. Only "otherwise lost" packets need to be rescued by optCA. Thus, in CA-STAR systems, the rate of input to an inlet (optCA buffer module)

would be less than the rate that new packets are generated at ordinary stations.

2. optCA learns from the control channel about the packets that have been transmitted by stations *prior* to their arrival. Scheduling is therefore done before the packets arrive to optCA.

3. The destinations of packets at buffer module Q_i are distributed over N-1 instead of N stations, when reflection is not applied[4].

6.4.2 Performance Measures

The following performance measures are used :

- *(normalised) throughput*, defined as the mean number of packets delivered per outlet per time slot; and
- *average packet delay*, defined as the average of the time intervals from when a new packet arrives at an inlet to when it is delivered to its destination outlet. In optCA-STAR networks, it is equivalent to the period from a packet's reception by optCA, to its transmission from optCA.

Also, let the relative efficiency of the FPCF algorithm, E(FPCF), be defined as the ratio of the (normalised) throughput of an IS using FPCF to the (normalised) throughput of the IS using the SDR algorithm, i.e.

$$E(FPCF) = \frac{Throughput(FPCF)}{Throughput(SDR)}. \tag{6.1}$$

6.4.3 Methodology

Throughput and delay characteristics of various $N \times B$ systems using 1) the FPCF algorithm and 2) the SDR algorithm for traffic assignment were obtained by simulating the steady-state behaviour of these systems, applying the methodology of quantitative stochastic simulation. All simulation results were obtained using AKAROA, [YAU96a]. Simulation runs were stopped when the steady-state estimates of all performance measures achieved or exceeded a given relative precision, at the 95% level of confidence level.

[4]Stations do not transmit to themselves

6.4.4 Results

Normalised throughput obtainable in ISs using the FPCF and SDR algorithms were first analysed for $N \times B$ = 10×10, 10×20, 10×40, and 10×100 systems.

Results for the 10×10 IS are presented in Table 6.1. The first column gives the values of p, the normalised offered load. The second column shows the point estimates of the throughput together with their 95% confidence intervals, when the FPCF algorithm was used for traffic assignment in the IS. The results obtained by the IS when SDR was used are contained in column three. The last column shows the estimate of the efficiency of FPCF, E(FPCF) as the ratio of the point estimates of the corresponding throughputs.

One can see from Table 6.1 that FPCF achieved near 100% efficiency for p between 0.1 and 0.9. At maximum offered load (p =1.0), the FPCF system gives 95.5% of the throughput achieved by SDR. Both algorithms yield a throughput greater than 99% of the offered traffic for values of p up to 0.8.

p	Thruoghput(FPCF)		Throughput(SDR)		E(FPCF)
1.0000	0.9025	(0.9018, 0.9032)	0.9450	(0.9441, 0.9460)	95.50%
0.9500	0.8870	(0.8863, 0.8877)	0.8976	(0.8969, 0.8982)	98.82%
0.9000	0.8644	(0.8637, 0.8651)	0.8693	(0.8685, 0.8702)	99.43%
0.8000	0.7929	(0.7922, 0.7935)	0.7947	(0.7941, 0.7953)	99.77%
0.7000	0.6994	(0.6989, 0.6999)	0.7000	(0.6995, 0.7005)	99.91%
0.6000	0.6002	(0.5996, 0.6007)	0.5995	(0.5989, 0.6000)	100.1%
0.5000	0.5005	(0.5000, 0.5010)	0.5005	(0.5000, 0.5010)	100.0%
0.4000	0.4002	(0.3999, 0.4005)	0.4002	(0.3999, 0.4005)	100.0%
0.3000	0.3001	(0.2998, 0.3003)	0.3000	(0.2998, 0.3002)	100.0%
0.2000	0.2000	(0.1998, 0.2002)	0.2000	(0.1998, 0.2002)	100.0%
0.1000	0.10000	(0.09991, 0.1001)	0.10000	(0.09991, 0.1001)	100.0%

Table 6.1: Normalised throughput of 10×10 interconnection systems using 1) the FPCF algorithm, and 2) the SDR algorithm for traffic assignment, as a function of p, the normalised offered load.

One can conclude that in this case, i.e. for B=10 and N=10, FPCF is as good as SDR. Furthermore, by resolving destination conflicts which could

limit the throughput to about 60%, either of these two assignment algorithms offers very good performance of the IS.

As the size of the inlet buffers are increased, the performance of both algorithms is expected to improve. The question is how does buffer sizes affect their *relative efficiency* ? A 10×20, 10×40, and 10×100 system was simulated in order to answer this question. Results are summarised in Table 6.2, Table 6.3, and Table 6.4 respectively.

p	Throughput(FPCF)		Throughput(SDR)		E(FPCF)
1.000	0.9517	(0.9510, 0.9525)	0.9732	(0.9724, 0.9740)	97.79%
0.9500	0.9311	(0.9304, 0.9317)	0.9237	(0.9232, 0.9243)	100.8%
0.9000	0.8958	(0.8953, 0.89640)	0.8884	(0.8876, 0.8891)	100.8%
0.8000	0.7993	(0.7986, 0.8000)	0.7993	(0.7986, 0.7999)	100.0%
0.7000	0.7001	(0.6996, 0.7006)	0.7001	(0.6996, 0.7006)	100.0%
0.6000	0.6002	(0.5997, 0.6008)	0.5995	(0.5989, 0.6000)	100.1%
0.5000	0.5005	(0.5000, 0.5010)	0.5005	(0.5000, 0.5010)	100.0%
0.4000	0.4002	(0.3999, 0.4005)	0.4002	(0.3999, 0.4005)	100.0%
0.3000	0.3001	(0.2998, 0.3003)	0.3000	(0.2998, 0.3002)	100.0%
0.2000	0.2000	(0.1998, 0.2002)	0.2000	(0.1998, 0.2002)	100.0%
0.1000	0.10000	(0.09991, 0.1001)	0.10000	(0.09991, 0.1001)	100.0%

Table 6.2: Normalised throughput of 10×20 interconnection systems using the FPCF algorithm, and the SDR algorithm for traffic assignment, as a function of p, the normalised offered load.

One can observe that for p =1.0, as B is increased from 20 to 100, the efficiency of FPCF relative to SDR is lifted from 97.75% to 100.0%. More interesting, results show that the FPCF algorithm yielded a throughput equal to or greater that of SDR for all other values of p considered, suggesting that FPCF could achieve "*super-optimal* " throughput. For instance, E(FPCF) was 100.8% when B =20, and 101% when B =100, at p =0.95. Moreover, an examination of the throughput of the FPCF and SDR systems at p=0.95, shows that the superiority of FPCF is statistically significant at the 0.95 level of confidence. The throughput of FPCF improves as B increases, as intuitively expected, because the "future planning window" of FPCF which equal B-1

p	Throughput(FPCF)		Throughput(SDR)		E(FPCF)
1.000	0.9781	(0.9773, 0.9789)	0.9858	(0.9852, 0.9865)	99.22%
0.9500	0.9476	(0.9468, 0.9485)	0.9361	(0.9353, 0.9370)	101.2%
0.9000	0.8997	(0.8988, 0.9006)	0.8946	(0.8939, 0.8953)	100.6%
0.8000	0.7994	(0.7987, 0.8000)	0.7994	(0.7987, 0.8000)	100.0%
0.7000	0.7001	(0.6996, 0.7006)	0.7001	(0.6996, 0.7006)	100.0%
0.6000	0.6002	(0.5997, 0.6008)	0.5995	(0.5989, 0.6000)	100.1%
0.5000	0.5005	(0.5000, 0.5010)	0.5005	(0.5000, 0.5010)	100.0%
0.4000	0.4002	(0.3999, 0.4005)	0.4002	(0.3999, 0.4005)	100.0%
0.3000	0.3001	(0.2998, 0.3003)	0.3000	(0.2998, 0.3002)	100.0%
0.2000	0.2000	(0.1998, 0.2002)	0.2000	(0.1998, 0.2002)	100.0%
0.1000	0.10000	(0.09991, 0.1001)	0.10000	(0.09991, 0.1001)	100.0%

Table 6.3: Normalised throughput of 10×40 interconnection systems using the FPCF algorithm, and the SDR algorithm for traffic assignment, as a function of p, the normalised offered load.

time slots increases too.

"Super-optimal" performance can be intuitively explained for FPCF. SDR gives optimal performance (100% assignment efficiency) in the sense that it optimises (maximises) the number of packets that could be transmitted conflict-free during one time-slot given the System State Matrix, S [CHEN91], [CHEN94]. However, under SDR (and other previously proposed algorithms), a new packet at an inlet i would always be buffered in buffer Q_i unless the buffer is full, in which case the new packet is discarded. Thus, packets are not screened at their arrival times if the corresponding buffers are not full. In contrast, FPCF uses Forward Planning of conflict-free transmission to selectively accept only those new packets which could be transmitted conflict-free during one of the future B-1 time-slots - instead of discarding new arrivals indiscriminately when full. Thus, buffer "real-estate" is more productively used, and the performance of the IS under FPCF can even surpass that when under SDR. If this level of performance could be verified for larger systems, then we could conclude that FPCF is an attractive alternative regardless of the size of the IS.

p	Throughput(FPCF)		Throughput(SDR)		E(FPCF)
1.0000	0.9915	(0.9906, 0.9924)	0.9915	(0.9907, 0.9924)	100.0%
0.9500	0.9500	(0.9495, 0.9506)	0.9406	(0.9398, 0.9415)	101.0%
0.9000	0.8999	(0.8993, 0.9004)	0.8954	(0.8946, 0.8963)	100.5%
0.8000	0.7994	(0.7987, 0.8000)	0.7994	(0.7987, 0.8000)	100.0%
0.7000	0.7001	(0.6996, 0.7006)	0.7001	(0.6996, 0.7006)	100.0%
0.6000	0.6002	(0.5997, 0.6008)	0.5995	(0.5989, 0.6000)	100.1%
0.5000	0.5005	(0.5000, 0.5010)	0.5005	(0.5000, 0.5010)	100.0%
0.4000	0.4002	(0.3999, 0.4005)	0.4002	(0.3999, 0.4005)	100.0%
0.3000	0.3001	(0.2998, 0.3003)	0.3000	(0.2998, 0.3002)	100.0%
0.2000	0.2000	(0.1998, 0.2002)	0.2000	(0.1998, 0.2002)	100.0%
0.1000	0.10000	(0.09991, 0.1001)	0.10000	(0.09991, 0.1001)	100.0%

Table 6.4: Normalised throughput of 10×100 interconnection systems using the FPCF algorithm, and the SDR algorithm for traffic assignment, as a function of p, the normalised offered load.

In order to compare the performance of FPCF with SDR for larger systems, their throughput as a function of system size (N) was studied for p =1.0, 0.95, and 0.9. Results are contained in Table 6.5, Table 6.6, and Table 6.7 respectively. Systems with N =3, 5, 10, 20, 30, 40, and 100 inlets/outlets were considered at each traffic level, assuming B =25. The results show that FPCF achieves similar or higher throughput than SDR, suggesting that FPCF remains a good solution as system size increases, *even if B is not increased.*

For applications of ISs in communication networks and switches, it is interesting to see the trade-off between mean delay and throughput. The delay-throughput characteristics of a 10×20, 10×40, and a 10×100 system are plotted in Fig. 6.2, 6.3, and 6.4 respectively. One can see that the difference in throughput-delay characteristics between FPCF and SDR are not significant. When B is small, FPCF is slightly inferior to SDR. For larger B, FPCF is slightly superior to SDR.

Net Size	Throughput(FPCF)		Throughput(SDR)		E(FPCF)
3.000	0.9834	(0.9795, 0.9873)	0.9755	(0.9718, 0.9792)	100.8%
5.000	0.9730	(0.9693, 0.9767)	0.9780	(0.9737, 0.9822)	99.49%
10.00	0.9633	(0.9596, 0.9669	0.9634	(0.9597, 0.9671)	100.0%
20.00	0.9527	(0.9499, 0.9554)	0.9522	(0.9486, 0.9557)	100.1%
30.00	0.9509	(0.9484, 0.9534)	0.9570	(0.9540, 0.9599)	99.36%
40.00	0.9532	(0.9518, 0.95470)	0.9485	(0.9447, 0.9522)	100.5%
100.0	0.9487	(0.9514, 0.05770)	0.9457	(0.9427, 0.9487)	100.3%

Table 6.5: Normalised throughput of $N\times25$ systems using the FPCF algorithm, and the SDR algorithm for traffic assignment, as a function of N. p=1.0.

6.4.5 Computational Complexity Comparison

When traffic assignment is applied in lightwave networks or electronic switches, the computational complexity of the traffic assignment algorithm can become a critical factor. An algorithm with high computational complexity could become the system's (electronic) bottleneck.

Let the *time computational complexity* (C_T) be defined as the maximum number of time steps needed for executing the algorithm in an $N \times B$ IS. Following [CHEN91], [CHEN92], and [CHEN94], we assume that an assignment, comparison, addition, or subtraction operation can be done in one time step.

During each slot, for each new packet, FPCF performs at most B -2 comparisons with elements in the S matrix (to find an empty location), B-1 comparisons with elements of matrix Δ (to see if the destination of the packet has already being assigned), 1 assignment to update S, and 1 assignment to update Δ. In the worst case N packets will need processing. Thus this gives the complexity of N (B -2 + B -1 + 2). Additionally, during the same slot FPCF performs 1 comparison and at most 2 assignments with V, and E. Finally, FPCF makes N assignments (to zero) to the Eth column of S and A. This gives C_T(FPCF)=N ($2B$ -1)+$2N$ +6. The performance of FPCF was shown to be largely unaffected as N increased, *even when B was constant.* Based of this, B will be treated as a constant. In fact the results showed that B=10

Net Size	Throughput(FPCF)		Throughput(SDR)		E(FPCCF)
3.000	0.9451	(0.9405, 0.9496)	0.9388	(0.9342, 0.9433)	101.0%
5.000	0.9429	(0.9394, 0.9465)	0.9321	(0.9285, 0.9357)	101.2%
10.00	0.9357	(0.9320, 0.9394)	0.9249	(0.9204, 0.9294)	101.2%
20.00	0.9298	(0.9257, 0.9338)	0.9260	(0.9229, 0.9290)	100.4%
30.00	0.9343	(0.9314, 0.9372)	0.9262	(0.9220, 0.9305)	100.9%
40.00	0.9338	(0.9320, 0.9357)	0.9229	(0.9189, 0.9269)	101.2%
100.0	0.9332	(0.9318, 0.9346)	0.9253	(0.9221, 0.9284)	100.9%

Table 6.6: Normalised throughput of $N\times25$ systems using the FPCF algorithm, and the SDR algorithm for traffic assignment, as a function of N. p=0.95.

already gave very satisfactory performance.

As mentioned, the high complexity of SDR has motivated the development of suboptimal but less complex algorithms: namely the MRS algorithm [CHEN91], [CHEN92], the K-HOL algorithm [CHEN94], and the RS algorithm [CHIP93], [CHEN94]. The worst case computational complexity of SDR and MRS were analysed in [CHEN91], and the results for K-HOL and RS were given in [CHEN94]. These results are collected in Table 6.8 As one can see, FPCF and K-HOL has considerably lower time computational complexity. The C_T of FPCF and K-HOL is of the order O(N), while that of SDR and RS are of the order O(N^4), and that of MRS is O(N^2).

Almost all operations of the FPCF algorithm are matrix operations with potential parallelism that can be exploited by execution on a simple vector or multiprocessor system. The execution time of FPCF can thus be further reduced. Also, many of the comparisons can be performed in parallel using associative testing. Thus, FPCF seems well positioned to benefit from the use of special hardware.

6.4.6 Comparison of Buffer Organisation Complexity

Physical buffer organisation is defined by the electronic components of inlet buffers, and the access paths they support. A typical buffer contains a mem-

Net Size	Throughput(FPCF)		Throughput(SDR)		E(FPCF)
3.000	0.8979	(0.8949, 0.9008)	0.8960	(0.8924, 0.8997)	100.2%
5.000	0.8968	(0.8932, 0.9005)	0.8870	(0.8826, 0.8914)	101.1%
10.00	0.8972	(0.8935, 0.9010)	0.8879	(0.8841, 0.8917)	101.1%
20.00	0.8921	(0.8879, 0.8962)	0.8856	(0.8817, 0.8894)	100.7%
30.00	0.8939	(0.8899, 0.8980)	0.8947	(0.8911, 0.8984)	99.91%
40.00	0.8947	(0.8916, 0.8978)	0.8918	(0.8875, 0.8962)	100.3%
100.0	0.8975	(0.8959, 0.8991)	0.8957	(0.8928, 0.8987)	100.2%

Table 6.7: Normalised throughput of $N\times25$ systems using the FPCF algorithm, and the SDR algorithm for traffic assignment, as a function of N. p=0.9.

ory, a time multiplexed read address/data bus, a write address/data bus, and several control and power lines. The memory is an array of addressable storage elements called words. For convenience, divide the memory into B packet size addressable locations, each of which is made of a fixed number of words. The packet size locations have unique addresses, and are identical in storage characteristics.

On top of this physical buffer organisation, we can impose a particular *logical buffer organisation (logical structure)*, that defines relationships between the packets stored in the physical buffer. Common logical buffer organisation of packets in ISs are the FIFO (First-In-First-Out) queue, LDF (Largest accumulated packet Delay served First) queue [CHLA91], and multi-queues (e.g. many algorithms require N FIFO queues of packets to be formed from one physical buffer).

A logical buffer structure is typically *created* using pointers which record the relationship between packets stored in a given physical buffer organisation. A logical buffer organisation is usually *maintained* by updating pointers when packets are added/transmitted to the physical buffer. Imposing a logical buffer organisation on top of a physical buffer organisation thus has resource (pointers) and processing time (updating pointers) overhead.

The logical structures that need to be maintained for each inlet buffer under FPCF, SDR, MRS, K-HOL, and RS are compared in Table 6.9. The

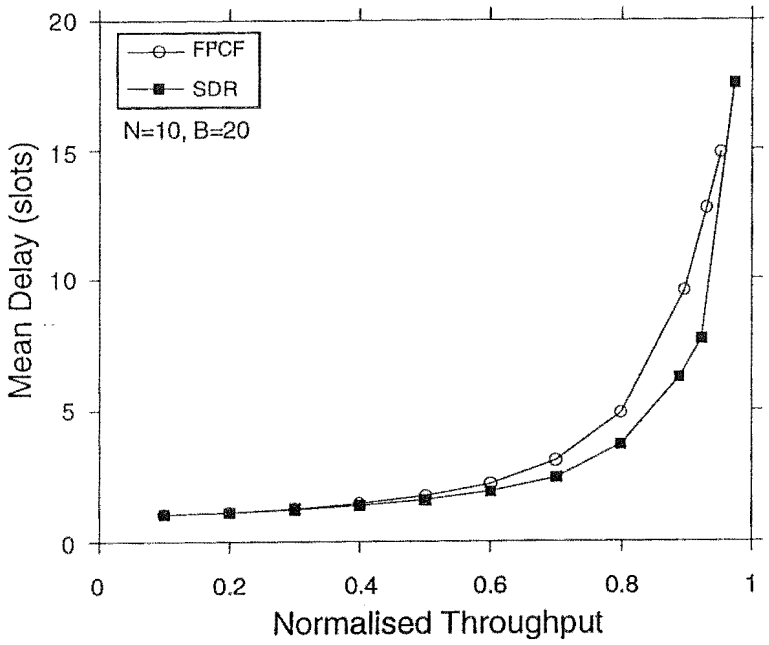


Figure 6.2: Throughput-delay characteristic of 10×20 systems using FPCF and SDR.

SDR and MRS algorithms need N FIFO queues of packets to be formed in each inlet buffer. In the case of networks using the RS algorithm [CHIP93], it was assumed that packets at each inlet (station) are organised in N FIFO queues using N separate FIFO buffers. Table 6.9 shows that FPCF has the simplest logical buffer organisation of the algorithms considered: no logical structures in inlet buffers need to be maintained.

This improvement from Forward Planning can be explained as follows. Previously developed algorithms try to find an assignment schedule which maximises the number of packets that can be transmitted conflict-free during a time slot, subject to R1) to R3). Let the schedule be specified by an $N \times N$ transmission matrix $X=[x_{i,j}]_{N\times N}$, such that $x_{i,j}=1$ if inlet i is scheduled to transmit a packet to outlet O_j; $x_{i,j}=0$ o.w.. X thus specifies all packet transfers from inlets to outlets for the next time slot.

The need for logical buffer organisation can be seen when actioning a transmission schedule. For each $[x_{i,j}]=1$ $(1\geq i,j \geq N)$, a search has to be conducted in inlet buffer Q_i for a packet that meets these conditions:-

1. the packet must be destined for outlet O_j, and
2. the packet must be the first to arrive to inlet i, of all those that are destined for O_j.

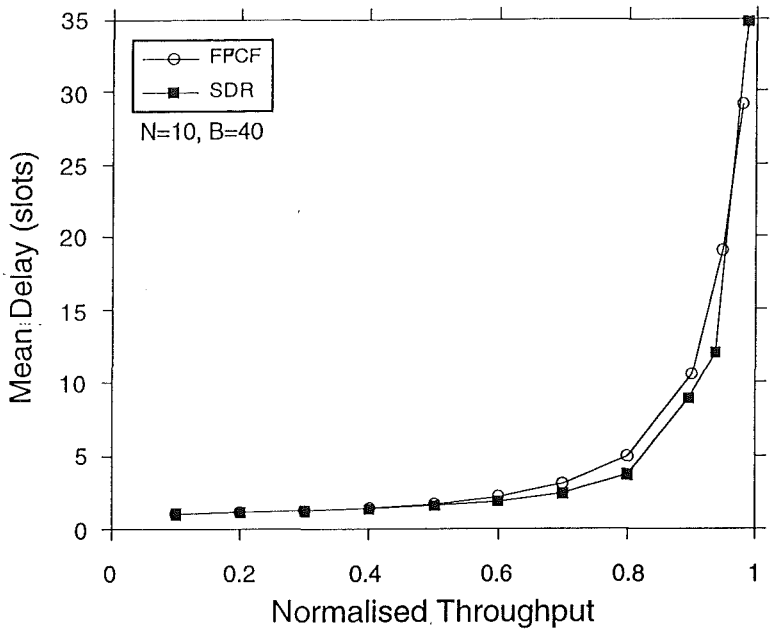


Figure 6.3: Throughput-delay characteristic of 10×40 systems using FPCF and SDR.

To support these searches, a logical buffer organisation has to be imposed on top of the physical buffer organisation. Typically, packets in each inlet buffer were assumed to be logically organised into N FIFO queues, one per possible destination. Then for each $[x_{i,j}]$=1, the packet to be transmitted can be *identified as the one at the head of the jth FIFO queue in the ith buffer.* Alternatively, each inlet can be equipped with N FIFO buffers, one per possible destination.

In contrast, FPCF uses the knowledge of future transmissions to streamline buffer organisation. Under FPCF, all packets that are scheduled for transmission during a time slot are simply identified by the Eth column of S. Namely, if $s_{iE} \neq 0$, then the packet in the E-th location of inlet buffer Q_i should be transmitted. The column index E rotates *deterministically* after each slot. Since the buffer locations of all packets that are scheduled for transmission are completely identified by the column index E (whose value deterministically increments by one, modulo B, after each time-slot), there is no need to search inlet buffers for the locations of the packets that are scheduled for transmission. Thus, no queues nor lists of packets need to be created nor maintained. No logical relationships between packets in a buffer needs to be recorded. Planning the future transmission of a packet becomes equivalent to determining the physical address for storing a packet. Logical buffer organisation is therefore completely subsumed by the Forwarding Planning of packet transmissions. Furthermore, the FIFO service order is naturally maintained,

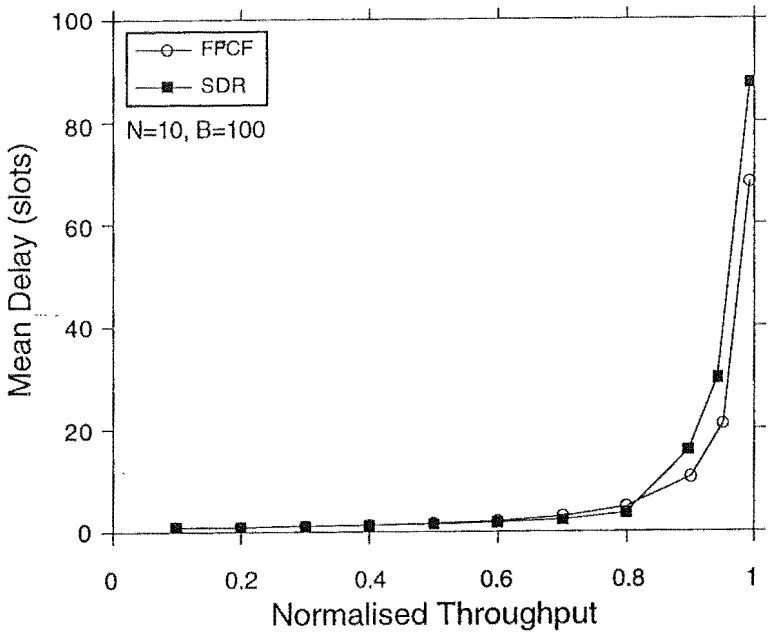


Figure 6.4: Throughput-delay characteristic of 10×100 systems using FPCF and SDR.

since new packets are scheduled when they arrive.

Finally, with FPCF, inlet buffers are accessed using relatively simple access modes. For input, random access (**R**) write is needed, but reading is done using column access mode (**C**) which permits somewhat simpler addressing.

6.5 Conclusions

The Forward Planning Conflict Free (FPCF) algorithm for traffic assignment in interconnection systems was proposed. FPCF differs from previously introduced algorithms in that it plans transmissions within an $N \times B$ IS for up to B-1 time slots in advance, instead of scheduling them just for the next time-slot.

The performance of FPCF was analysed and compared with the SDR algorithm, which offers the best performance from among other previously proposed algorithms. Additionally, the computational complexity of FPCF was compared with that of SDR and such approximate algorithms as MRS, K-HOL and RS, proposed for their lower computational complexity than SDR. As shown, FPCF offers similar or even better throughput than SDR. Performance better than under SDR can be achieved, since the Forward Planning of packet transmissions is used to determine whether incoming packets can be

ALGORITHM	Computational Complexity
SDR	$\frac{2}{3}N^4+\frac{1}{6}N^3+\frac{4}{3}N^2+\frac{5}{3}N$
MRS	$12N^2-2N$
K-HOL	$(4K$ -2$)N$ + $2K$ -2
RS	$\sum_{i=1}^{N}\sum_{j=1}^{N}(N-i+2)(N-j+2)(N+i-1)$
FPCF	$(2B$ -1$)N$ +$2N$ +6

Table 6.8: Worst case time computational complexity (C_T) of various traffic assignment algorithms.

Algorithm	Buffer access modes		Logical Structures Maintained
	Read	Write	
SDR	R	R	N FIFO Queues per inlet buffer
MSR	R	R	N FIFO Queues per inlet buffer
K-HOL	R	R	1 FIFO Queue per inlet
RS	R	R	N Separate FIFO buffers per inlet
FPCF	C	R	None required

Table 6.9: Buffer access modes, and buffer organisation needed by various algorithms.

transmitted conflict-free in the foreseeable future, rejecting in advance those of them for which a destination conflict within the next B-1 slots would be unavoidable.

Importantly, FPCF was shown to have drastically lower computational complexity than SDR (O(N) compared with O(N^4)), and the same or lower order of complexity than the various approximate algorithms. This is due to the fact that with FPCF, only packets arriving during the current time-slot

need to be scheduled - existing packets are already scheduled for transmission during (one of) the future time slots. Moreover, FPCF can use the knowledge of future transmissions to streamline buffer organisation. Unlike previous solutions, no logical relationships between packets in a buffer (i.e. no FIFO queues, Largest Delay First queues, lists of packets, nor any logical structure) needs to be recorded. This saves resources (for logical buffer organisation) and processing time (for maintaining logical structures). FPCF has the potential for further speedup using SIMD, MIMD, or associative testing hardware. These results suggest that FPCF is an attractive option for conflict-free transmission scheduling in applications such as optCA-STAR networks.

Chapter 7

FPCF based optCA-STAR Protocols

The reason for the introduction of optCA-STAR protocols based on the FPCF algorithm is that the number of operations which need to be executed during a time slot increases very rapidly with system size, when optCA operates according to the optCA-MRS* protocol. In particular, the time computational complexity of optCA-MRS* grows to the order of $O(N^2)$, as shown in Chapter 5. Additionally, optCA-MRS* requires complex logical buffer organisation. Under optCA-MRS*, packets stored in each optCA buffer module has to be logically organised into N-1 FIFO queues, one queue per possible destination of the packets. As N grows, the (electronic processing) operations for protocol execution, and for maintaining N-1 FIFO queues per buffer module, may themselves become the network's electronic bottleneck.

The FPCF conflict-free scheduling algorithm was shown in Chapter 6 to have significantly lower computational complexity ($O(N)$) than the algorithms previously used in "request-schedule-then-transmit" WDM networks. Simultaneously, FPCF yielded equal or higher efficiency, and enjoys extremely low logical buffer organisation complexity. By integrating the FPCF algorithm into the MAC protocol of optCA, one expects a reduction in the protocol's processing complexity and a reduction in buffer organisation overhead relative to optCA-MRS*, similar to that yielded by FPCF over MRS.

The extensions to FPCF for serving the MAC functions of optCA are briefly discussed next, followed by definitions of the actual FPCF optCA-STAR protocols. The improvements obtainable with the FPCF based networks are then assessed in section 7.3. As mentioned, instead of requiring optCA to receive "otherwise lost" packets, and then storing them into an

electronic memory and then retransmit them later, the "otherwise lost" packets can be optically buffered by CA until their destinations are free to receive them, thereby obviating the need for data receivers, data transmitters, and electronic memory. The design of an all-optical optCA-STAR network is proposed in section 7.4.1, assuming the availability of wavelength converters.

7.1 optCA-STAR Protocols Based on the FPCF Algorithm

The basic FPCF algorithm studied in Chapter 6 accepts as input an array of destination addresses (the H matrix) of incoming packets, and schedules the transmission of these packets during one of the B-1 future time slots, so that their destinations would be free to receive them on their arrival. The extension of FPCF for optCA MAC implies the added functions of planning packet receptions (deciding which packets need "rescuing", prior to their arrival to optCA), and planning mini-slot transmissions. For some computational savings, we will integrate packet reception and mini-slot transmission planning with FPCF scheduling. FPCF can also be integrated with a Reflection mechanism to guarantee the delivery of packets. First of all, an optCA-STAR protocol based on the FPCF algorithm, called optCA-FPCF/B, will be described. Then one incorporating a Reflection procedure, called the optCA-FPCF/R, will be presented.

Following the naming convention introduced in chapter 1, we shall refer to optCA-STAR networks operating according to the optCA-FPCF/B protocol as "optCA-FPCF/B networks". Similarly, we shall refer to optCA-STAR networks operating according to the optCA-FPCF/R protocol as "optCA-FPCF/R networks". The structure of the optCA-FPCF/B and optCA-FPCF/R protocols are summarised in Fig. 7.1.

7.2 The optCA-FPCF/B Protocol

Ordinary stations of an optCA-STAR network are freed from the burden of destination conflict resolution, so their MAC protocol could be simplified. Thus, the optCA-FPCF/B protocol is defined by i) the protocol for ordinary stations, and ii) the protocol for optCA, as shown in Fig. 7.1.

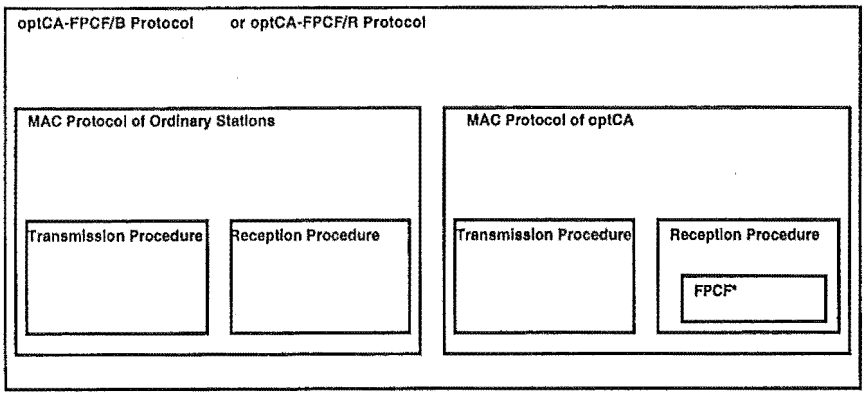


Figure 7.1: Structure of the optCA-FPCF/B or optCA-FPCF/R protocols.

7.2.1 MAC Protocol of Ordinary Stations

The optCA-FPCF/B MAC protocol for ordinary network stations is the same as that specified for optCA-MRS*, see section 5.2.2 on page 115.

7.2.2 MAC Protocol for the optCA Station

Let each of the N buffer modules of optCA be sized to store B packets, $B \geq 2$. Think of the N optCA buffers as a matrix $C=[c_{ij}]_{N\times B}$ of packet size memory locations, see Fig. 5.4. Since each row of C represents B *memory locations* of one buffer, the following access rules must be observed during one time slot: (i) there can be at most one "write" (receive) and one "read" (transmit) operation per row, and (ii) a read and a write operation on the same row must be on distinct locations.

Let optCA maintain the following variables[1]

- Planned Reception Matrix, $P = [p_i\ ]_{N\times 1}$. At the beginning of t, $p_i = k$ if during[2] t, optCA should receive the packet transmitted by S_i, storing it in location k of the ith buffer module; $p_i = 0$ o.w..

[1]Normal scope rules apply. Thus variables declared for the definition of the optCA MAC protocol algorithm are visible only within its definition. A variable with a given name is as declared in the protocol specification, even if the same name have been used to identify another variable in another protocol. Normal precedence rules also apply. Variables and notation declared outside the definition of a protocol are visible within its definition, so long as they have not being redefined in the protocol specification.

[2]Recall from section 4.2.1 that t denotes the t-th time slot.

- Next Reception Matrix, $P^+ = [p_i^+]_{N\times 1}$. At the beginning of t, $p_i^+ = k$ if during $t+1$ optCA should receive the packet from S_i into the location k ; $p_i^+ = 0$ o.w..
- Mini-slot Transmission Matrix, $M = [m_i\ ]_{N\times 1}$. m_i is the channel index (number) which would be transmitted by optCA on the $N+i$th mini-slot during the current time slot.
- Enabled Column Counter, E. E is initialised to B during network initialisation, and decremented by one after each time slot; if E==0 then E is reset to B .

Procedure optCA Transmission (Executed during each slot)
Begin
CoBegin
for i=1, ..., N **do** Transmit m_i on the $N+i$th mini-slot of the outgoing control slot ;
forall c_{iE}, $i = 1, ..., N$ **doparallel**
if (c_{iE} is not empty) **then** transmit the packet in c_{iE} ;
CoEnd
$E = E$ - 1 ; **if** $(E == 0)$ **then** $E = B$;
End

Procedure optCA Reception (Executed during each slot)
CoBegin
forall $i = 1, ..., N$ doparallel
if $(p_i \neq 0)$ **then** receive the packet transmitted by S_i into c_{i,p_i} ;
receive mini-slots 1 to N ; $P = P^+$; update M and P^+ using the FPCF* algorithm ;
CoEnd

The elements of M in the Transmission procedure refer to the values they assumed prior to their update by FPCF* in the Reception procedure. The FPCF* procedure is used to plan transmissions and receptions over one time slot in advance, so reception decisions initially written in P^+ have to be assigned to P after each slot. Next, P^+ can be overwritten when FPCF* is invoked again, before the arrival of the packets planned for reception, see Fig. 7.2. For fast execution, P^+ and P can simply be rotated during every time slot to eliminate the need for the "P=P^+ " assignment. The timing of the control channel operations of optCA and FPCF* execution is specified in Fig. 7.2. Note that the buffer location occupied by a rescued packet can be re-used as soon as the packet has been transmitted from optCA. The reason is that packets transmitted from optCA will always be received.

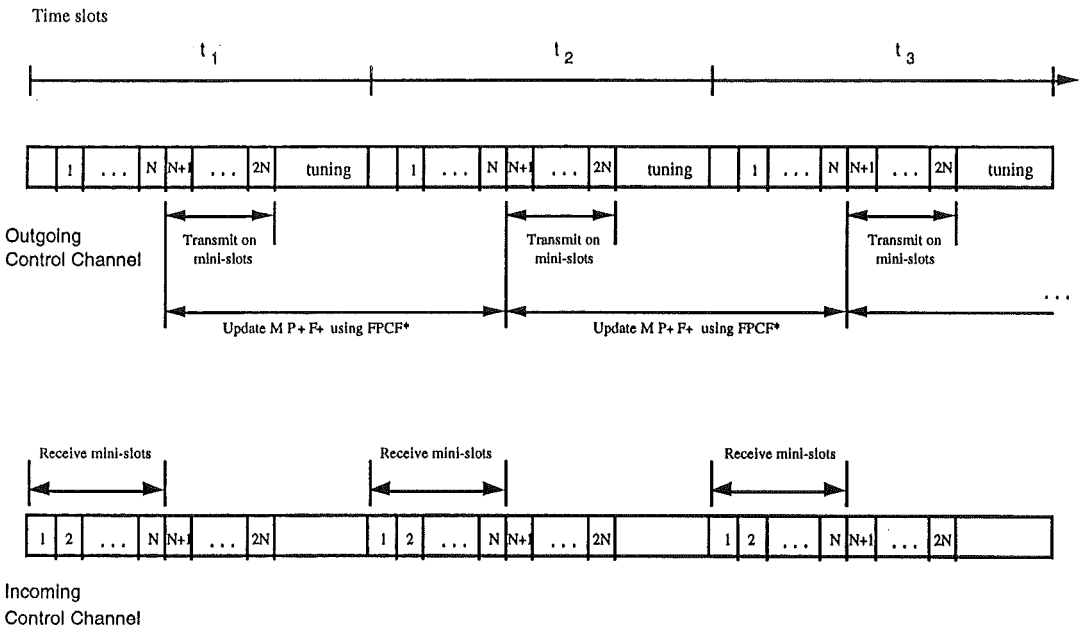


Figure 7.2: Timing of control channel operations and FPCF* execution according to the optCA-FPCF/B protocol.

7.2.3 FPCF* Algorithm

The FPCF* algorithm is used for planing the reception of "otherwise lost" packets into optCA, and for scheduling received packets for conflict-free transmission to their destinations.

FPCF* is invoked during t to plan packet receptions during t+2, and to schedule the transmission of those packets during one of the future B-1 time slots. optCA can plan receptions two slots in advance, since the control mini-slots it receives during t correspond to packets arriving during t+2 (see the Transmission procedure of ordinary stations). optCA has to start planning transmissions two time slots in advance since it needs (i) one time-slot to execute the FPCF* algorithm, and then (ii) notify destination stations of which channels to receive from one "tuning-period" in advance, using mini-slots N+1, ..., $2N$.

FPCF* uses four additional variables. Some of them are redundant because their values could be derived from the other variables. Their alternative representation helps reduce computational complexity.

- Future C Status Matrix[3], $F = [f_{ij}]_{N \times B}$. At the end of the FPCF-

[3]We avoid using the symbol C^+, since C refers to the N physical buffer memories,

Processing-time-period of t (Fig. 7.2), f_{ij} = destination address of the packet that will be in c_{ij} during t+2. If c_{ij} will be empty during t+2 then $f_{ij} = 0$.

- Destination Allocation Matrix, $\Delta = [\Delta_{ij}]_{N \times B}$. During t, $\Delta_{ij} = 1$ if one of f_{1j}, f_{2j}, ..., f_{Nj} equals i. $\Delta_{ij} = 0$ o.w..
- FaVoured Station (V) Counter. Initialised to 1 during network startup.
- Future E counter, E^+. E^+ is the value of E two slots from now. E+ is initialised to B-2.

Let $H = [h_1, h_2, ..., h_N]$ be the vector of N addresses contained in the first N mini-slots of the current control slot.

procedure FPCF*(input H) // Compute new P^+ and M.
Begin
$V = V + 1$; **if** $(V > N)$ **then** $V = 1$;
$E^+ = E^+ - 1$; **if** $(E^+ == 0)$ **then** $E^+ = B$;
for each S_i $i = V, V+1, ..., N, 1, 2, ..., V-1$ **do**
{ **if** $h_i \neq 0$ **then** // S_i transmitted a packet
if $\Delta_{h_i,E^+} == 1$ **then** // packet's destination already allocated to another packet
// S_i 's packet will not be received by its intended destina
{ find the first j s.t. ($f_{ij} == 0$ **AND** $\Delta_{h_i,j} == 0$), searching in the order
given by j = $E^+ - 1, ..., 1, B, B-1, ..., E^+ + 1$
if found, **then** $f_{ij} = h_i$; $\Delta_{h_i,j} = 1$; $p_i^+ = j$;
else $p_i^+ = 0$; // can't transmit packet during the next B-
} // time slots. Packet will be lost (!)
else // S_i 's packet will be successfully received. No need for optCA to attend
{ $p_i^+ = 0$; $\Delta_{h_i,E^+} = 1$; $m_{h_i} = i$; }
if ($f_{i,E^+} > 0$) **then** $m_{f_{i,E^+}} = N+i$;
} // endfor
$[f_{1,E^+}, f_{2,E^+}, ..., f_{N,E^+}] = [0, 0, ..., 0]$; $[\Delta_{1,E^+}, \Delta_{2,E^+}, ..., \Delta_{N,E^+}] = [0, 0, ..., 0]$;
End;

The role of counter V is to rotate the order in which stations are processed during each slot. The sooner a station is processed, the greater the probability that its packet could be buffered by optCA, if its packet needed buffering. Rotating the processing order ensures that stations are fairly treated. Let the above optCA-STAR protocol be named optCA-FPCF/B.

whereas the future status matrix is a logical description of an attribute of the future status of C.

As one can see, FPCF* is extended from the FPCF algorithm, which was shown to have lower computational complexity than previously proposed conflict-free scheduling algorithms. Also, through the use of forward planning, FPCF was found to deliver a similar or even higher throughput than the SDR algorithm which has the best performance of the previously proposed conflict-free scheduling algorithms. By using FPCF*, an improvement in computational complexity and performance is therefore expected.

In "request-schedule-then-transmit" networks the packet that is scheduled for transmission from a station during a time-slot is found as follows. It is the one that is destined for a specific destination station specified by the transmission schedule, and is the first-to-arrive of all of the packets in the station's transmit buffer that are destined for that station. To facilitate the search for the first-to-arrive-to-buffer packet destined for a specific station, packets in the transmit buffer of all stations are usually be organised into N-1 logical FIFO queues. Likewise, according to the optCA-MRS* protocol, packets in the buffer modules of optCA are logically organised into N-1 FIFO queues.

In contrast, according to FPCF* the packets in optCA that are scheduled for transmission during a time slot are identified by the E-th column of C. Specifically, if $c_{i,E} \neq empty$ then the packet stored in $c_{i,E}$ would be transmitted. The column index E is rotated after each slot. Thus the planning of future transmissions becomes equivalent to the planning of which location to store a rescued packet. Consequently, logical buffer organisation is now completely subsumed by Forwarding Planning of packet transmissions. No queues nor lists of packets need to be created or maintained. No logical relationships between packets in a buffer need to be recorded. FIFO service order is naturally maintained, since the scheduling of packet transmissions take place when the packets arrive to optCA.

7.2.4 Providing Delivery Guarantees

A packet p_i transmitted by S_i would be lost, if its destination would not be able to receive it during its original time of arrival, and optCA cannot find a future arrive time (within up to B slots from its original time of arrival) when it can be received conflict-free. This raises the question of whether it is worthwhile to provide a facility to eliminate any losses of packets. If the probability of packet losses is very low when the offered load is not too high, and if the major bandwidth consumers in the network are delay sensitive but can accept some packet loss provided that the probability of packet loss is below a specified

level, then such a facility may have little value. Connection acceptance or buffer congestion control [YAU92b], [YAU92c] maybe one alternative.

On the other hand, an advantage of a central placement of the conflict resolution function in CA-STAR networks is that it is very simple to eliminate any loss of packets. The E-th column of occupants of C (buffer modules of optCA) are transmitted during every time-slot. Hence, during every time slot there is at least one empty buffer location in each buffer module: namely the $(E+1)$-th location. This enables the optCA-FPCF/B protocol to be extended to eliminate any loss of packets using the idea of Reflection, first applied in the context of sCA-STAR networks in Chapter 4. Let e_{ik} be a pointer to the first empty location in the ith buffer encountered when searching for a suitable location to buffer p_i, following FPCF*.

To apply Reflection, if FPCF* cannot find a suitable location for buffering packet p_i, then optCA stores p_i into location e_{ik}, instead of loosing it. However, another packet in the kth column of C would have the same destination as that of p_i, so optCA assigns a *surrogate destination* for p_i, say station S_s. S_s should not be the destination of any packet in the kth column. An S_s for p_i is found by searching the k th column of the Δ matrix for an s s.t. $\Delta_{s,k}$==0, starting from row i+1, i+2, ..., circling back to row 1 to i -1 if necessary. A surrogate destination S_s is guaranteed to be found. Once found, assign f_{ik} =s, and $\Delta_{s,k}$ =1. Thus the transmit procedure will know where to transmit p_i, and will inform S_s that it should receive the packet, one slot in advance, using the $(N+s)$th mini-slot. The fact that S_s has been allocated for receiving p_i is noted in Δ.

If a station receives a packet that is not destined for it[4] but to S_d, then the station: (1) blocks the arrival of any new packets from its LLC layer during the next time-slot, and (2) transmits the packet in the usual way, i.e. treating it as its own packet addressed to S_d. Let us name the optCA-FPCF/B protocol enhanced with Reflection as optCA-FPCF/R.

Note that once a station has been chosen to serve as a surrogate during a slot, the station cannot receive another packet during that slot. For this reason, the time when surrogates are chosen affects performance. Assignment on arrival (immediate surrogate binding) may tie stations to be surrogates during a future slot, even if the station could have received a packet (subsequent rescued by optCA) destined for that station. We consider here the case where the selection of S_s is done just two time slots prior to the transmission

[4]One bit may be reserved in each mini-slot to serve as a Packet-Reflected indicator. optCA sets this bit in the mini-slot corresponding to a packet it plans to Reflect. Ordinary can identify Reflected packets in advance by testing just one bit.

of the packets in the kth column of C. This assigning of surrogates at the last possible moment is called Late Surrogate Binding (LSB), and has been shown to improve performance over immediate assignment versions. One can formally verify that Reflection is safe (packets will never be lost) under all traffic patterns and all network configurations, using a construction similar to that developed in Chapter 4.

7.3 Performance Evaluation

Results for the steady-state throughput and mean packet delay were obtained by means of quantitative stochastic simulation, as outlined in section 4.2.7 and described greater detail in the technical report [YAU96a]. For comparable results, we assumed the same modelling assumptions as those used in the analysis of previous optCA-STAR networks, and in the WDM networks studied in [CHEN91], [PAPA92] [CHLA91], [HUMB93], [CHEN92], [CHEN90], and [CHIP93], see section 4.2.7.

First, consider an optCA-FPCF/B network with N=10 stations, where all stations are a=5 slots from optCA.

7.3.1 Effect of Buffer Size on Throughput and Delay Characteristics

1) optCA-FPCF/B : Fig. 7.3 shows results for the normalised throughput as a function of offered load, q, for a=5, N=10, and varying B[5]. Throughput almost equal the offered load, for offered loads up to 96%. The mean packet delay (Fig. 7.4) is close to minimum (12 slots), regardless of B, provided the offered load is below 80%. By *locating* the destination conflict resolution function at just one station (the optCA) which resolve conflicts whilst packets are *en route* to their destinations, the optCA-FPCF/B network provides both near minimum packet delay and optimal throughput. The low mean packet delay is expected, since stations may transmit ready packets without waiting a long destination conflict resolution period, and yet even if a packet is involved in a destination conflict, it will not experience additional propagation delay. In fact, its arrival time would be re-scheduled (by buffering at optCA) so that it arrives to its destination as soon as its destination is free to receive it. The high throughput can be intuitively explained for optCA-FPCF/B since

[5]Recall that in optCA-STAR, B equals to the memory capacity (in packets) of each buffer module of optCA

the FPCF algorithm used by optCA (for re-scheduling the arrival times of packets so that they arrive to their destinations as soon as possible, subject to the conflict free and ordering constraints) is known to have high efficiency. Also, multiple re-transmissions of a packet is unnecessary.

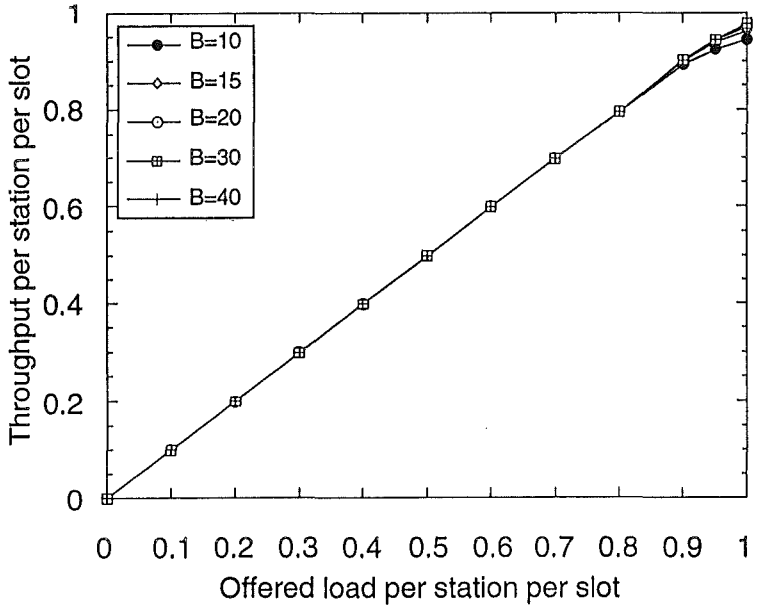


Figure 7.3: Throughput of optCA-FPCF/B versus load, for varying buffer sizes. N=10, a=5. Relative precision $\leq$5%

2) optCA-FPCF/R Protocol: Throughput and average packet delay of optCA-STAR networks using optCA-FPCF/R are shown in Fig. 7.5 and 7.6, respectively. optCA-FPCF/R also yields near optimal performance when the offered load is between 0 and 0.98. Above q=0.98, the throughput drops from 98% when q =0.98 down to about 88% (B =40) or 64% (B =10). This result was intuitively expected, since above q=0.98 the probability of applying Reflection becomes significant, whereas Reflection is a rare event at lower load levels. Fig. 7.5 and 7.6 show that the overhead of Reflection is significant only if the offered load is extremely high (over 0.98), and its effect is bounded.

7.3.2 Impact of Increasing Network Size:

1) optCA-FPCF/B: Fig. 7.7 shows that increasing network size from N =10 to 100 stations does not affect its efficiency. Increasing N from 3 to 5 stations increases the average packet delay somewhat at medium traffic, see Fig. 7.8, but further increases has little effect. These results demonstrate that optCA-FPCF/B remains a good solution as the network size increases, *even when B*

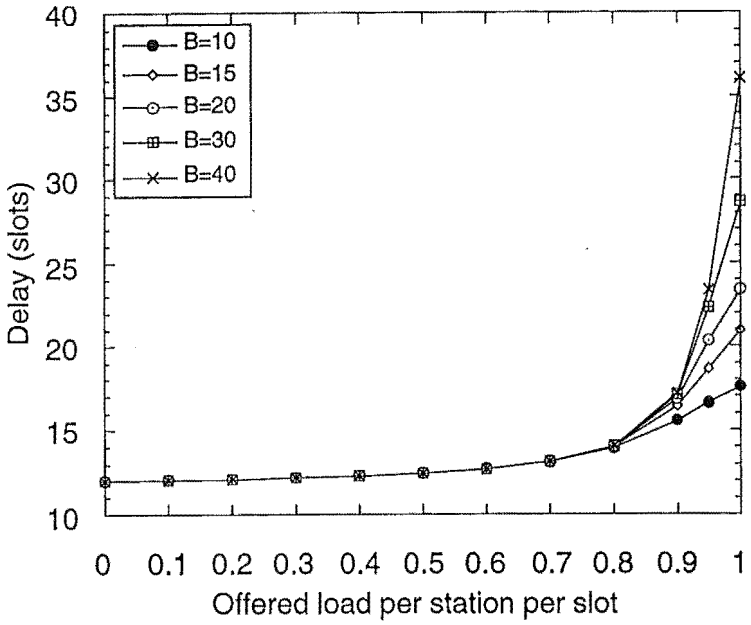


Figure 7.4: Mean packet delay of optCA-FPCF/B versus load, for varying buffer sizes. N=10, a=5. Relative precision $\leq$5%

remains constant.

2) optCA-FPCF/R: Fig. 7.9 and 7.10 reports the performance of optCA-FPCF/R under the same range of q and N. The results show that optCA-FPCF/R also enjoys near optimal performance, irrespective of the network size, when the offered load is between 0 and 0.98, *even when B remains constant*. Above q =0.98, there is some degradation in throughput and delay, irrespective of N, as expected.

7.3.3 Impact of Reflection on Performance

As a third step, the cost of employing Reflection is quantified. While protecting packets from being lost, Reflection has three overheads. First, when a packet is admitted by optCA for Reflection (instead of being lost), it will occupy one location of a buffer module of optCA for up to B -1 time slots. Thus a "to-be-reflected" packet increases buffer occupancy, increasing the probability that packets transmitted after it also needs to be reflected. Secondly, a reflected packet p_r is transmitted by optCA to a surrogate station which receives and re-transmits it. If another packet p_i transmitted by an ordinary station is destined for the surrogate station during the same time slot, then p_i would be involved in a destination conflict with p_r. So p_i would need buffering by optCA. This further increases the probability that packets following a

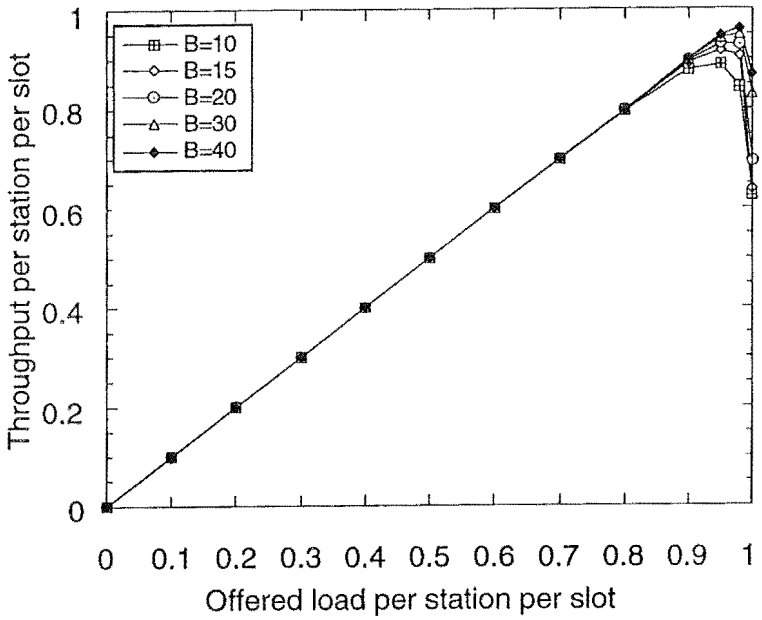


Figure 7.5: Throughput of optCA-FPCF/R versus load, for varying buffer sizes. N=10, a=5. Relative precision $\leq$5%.

reflected one would need to be reflected. Finally, a surrogate station receiving a reflected packet is blocked from accepting a new packet from its LLC during the next time-slot, since it must retransmit the reflected packet. This lowers throughput. The first two overheads have multiplier effects. Define *Reflection Multiplier factor,* $R_m\ (q\ )$, as

$$R_m\ (q\ ) = \frac{Prob(packet\ reflected \mid q)}{Prob(packet\ lost \mid q)}$$

where Prob(packet lost | q) denotes the probability that a packet from a given station is lost, when the network operates following optCA-FPCF/B, at load q. Similarly, Prob(packet reflected | q) denotes the probability that a packet from a given station is reflected[6], when the network operates following optCA-FPCF/R, at load q .

If the reflection of a packet does not increase the probability that later packets need reflecting, then R_m (q) = 1, for all q. If there is a multiplier effect, then $R_m\ (q)$> 1. Estimates of Prob(packet lost | q), Prob(packet reflected | q), and $R_m\ (q)$, for an optCA-STAR network with N=10 stations and B =40, are contained in Table 7.1. An examination of Table 7.1 suggests that Reflection does have a significant multiplier effect when the offered load is high. At the highest possible traffic level (q=1.0), $\widehat{R_m}(1.0) \approx 8$. The

[6]Irrespective of whether the packet is a "new" one, or one which has been reflected before.

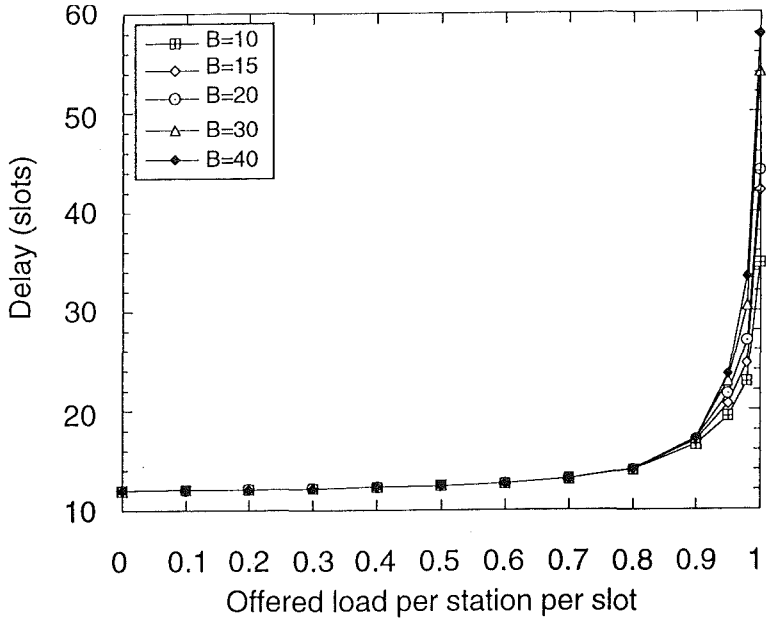


Figure 7.6: Mean packet delay of optCA-FPCF/R versus load, for varying buffer sizes. N=10, a=5. Relative precision $\leq$5%.

interpretation is that when optCA buffers one packet for Reflection (instead of loosing it), on average it creates the need to buffer 7 other packets for Reflection. Fortunately, the results show that the effect decreases to about $\widehat{R_m}(q) \approx 3$ for $q \leq 0.99$. This suggests that the overhead of Reflection is low except under maximum load. Moreover, the probability that Reflection occurs in optCA-FPCF/R diminishes quickly, from 0.13 at maximum load, to 0.0060 at q=0.96. This explains the near optimal performance of optCA-FPCF/R provided the offered load is not too high.

7.3.4 Analysis of Computational Complexity

As previously defined, let the *network computational complexity* of protocol A (C_N(A)) be the maximum number of scalar operations for MAC purposes that is performed in the network during one time slot, and the *time computational complexity* of protocol A (C_T(A)) be the maximum number of scalar operations for MAC purposes performed during one time slot by the station which has the most complex MAC procedure. Following [CHEN91], [CHEN94], an assignment, comparison, addition, or subtraction will be considered as a scalar operation. C_N can, for example, contrast the relative cost of the electronic processing elements required. If we are concerned about the MAC protocol's electronic processing operations creating a bottleneck, then C_T may be a more

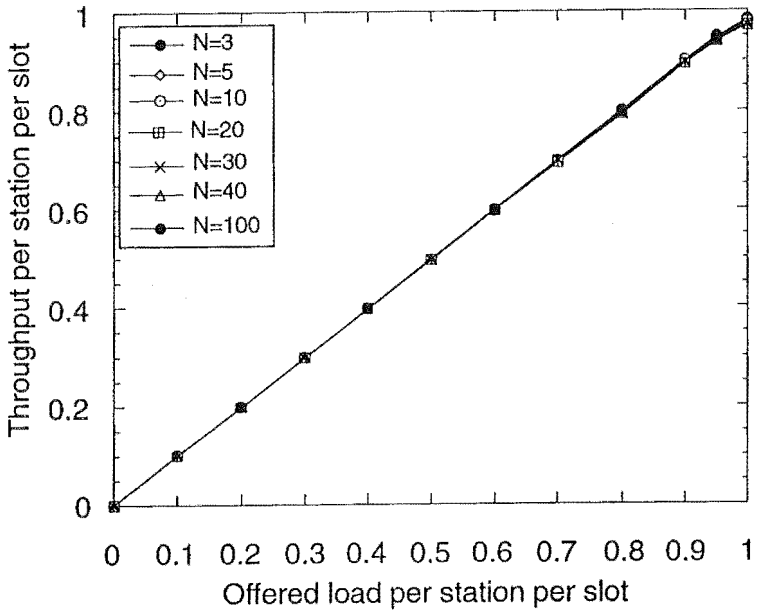


Figure 7.7: Throughput of optCA-FPCF/B versus load, for varying number of stations. B=25, a=5. Relative precision $\leq$5%.

suitable complexity index. We now derive the expressions for the C_T and C_N of optCA-STAR when operating under the optCA-FPCF/R protocol.

$$C_T(optCA-FPCF/R) = C_T(transmission\ procedure\ of\ optCA-FPCF/R) + C_T(reception\ procedure\ of\ optCA-FPCF/R). \quad (7.1)$$

During one time-slot, the transmission procedure of optCA-FPCF/R makes at most N comparisons (with c_{iE}), 1 subtraction (to E), 1 comparison (of E with zero), and 1 assignment (to E), so

$$C_T(transmission\ procedure\ of\ optCA-FPCF/R) = N+3. \quad (7.2)$$

During one time-slot, the reception procedure of optCA-FPCF/R makes at most N comparisons (of p_i with zero). In addition, it invokes the FPCF* algorithm for updating M and P^+.

During every slot, for each packet that will require buffering, FPCF* performs at most B-2 comparisons (with the F matrix) to find an empty location, B -1 comparisons (with the Δ matrix) to see if the destination of the packet

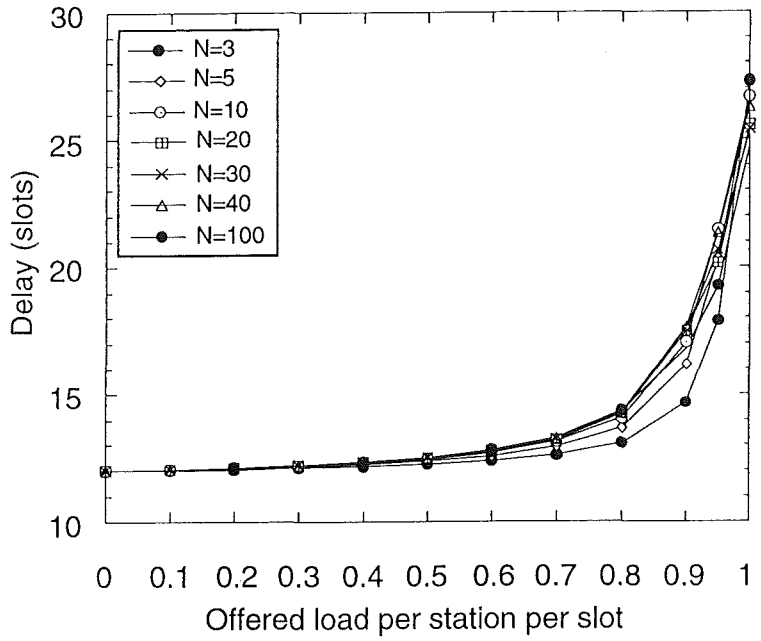


Figure 7.8: Mean packet delay of optCA-FPCF/B versus load, for varying number of stations. B=25, a=5. Relative precision $\leq$5%.

has already been assigned, 1 assignment to update F, 1 assignment to update Δ, and 1 assignment to e_{ij}. In the worst case N packets will need to be received by optCA. Also during one time-slot, FPCF* must scan the E^+th column of Δ to find zero entries (since the row indices of zero entries indicate stations that could serve as surrogates to any to-be-reflected packets in the corresponding column of C). This costs N comparisons (test for zero) and N -1 assignments (to F of destination addresses for to-be-reflected packets). Lastly during one slot, FPCF* makes at most N comparisons and N assignments to update M.

Adding gives

$$\begin{aligned} C_T(optCA-FPCF/R) &= N+3+N+(2BN+4N-1) \\ &= 6N+2BN+2. \end{aligned} \tag{7.3}$$

Next let us consider C_N of optCA-FPCF/R. During each time-slot, each ordinary station executes a station Transmission procedure (2 comparisons) and a Reception procedure (one comparison, and one assignment). Due to the use of Reflection, the station must also compare the address of each packet it receives (one comparison) to determine whether to accept or to re-transmit the packet. Thus,

$$C_T(Station\ MAC\ protocol\ of\ optCA-MRS^*)=5. \tag{7.4}$$

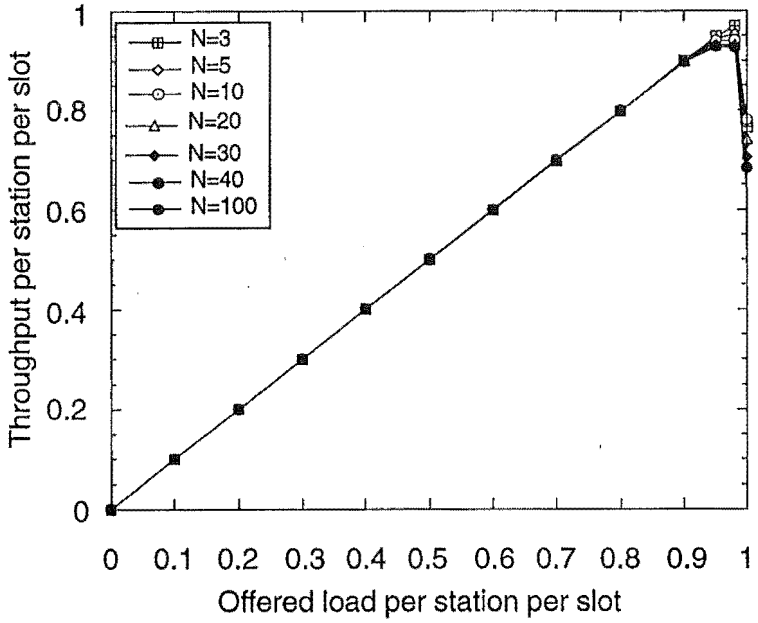


Figure 7.9: Throughput of optCA-FPCF/R* versus load, for varying number of stations. B=25, a=5. Relative precision $\leq$5%.

Adding the operations of the optCA station gives

$$C_N(optCA - FPCF/R) = 11N + 2BN + 2. \tag{7.5}$$

Table 9.5 compares C_N and C_T electronic processing complexities of the optCA-FPCF/B and optCA-FPCF/R networks with previous CA-STAR networks, and with that of networks using the request-schedule-then-transmit or the detect-and-retransmit-if-lost schemes. The comparison is based on a network with N stations, each with a transmit buffer of size B. As shown in column three, optCA-FPCF/B(/R) enjoys considerably lower C_N. The optCA-STAR networks obtains low C_N by complexity inversion: MAC tasks that burden all stations in the other architectures (e.g. deciding which packet to receive) are performed by just one station, i.e. by optCA. In this way, much replication of electronic processing in previous architectures has been eliminated. Ordinary stations need to process only one mini-slot during every time slot. In contrast, previous solutions [CHEN90], [CHEN91], [CHEN92], [CHLA91], [CHLA94], require all stations to process information in all mini-slots during every time-slot.

Expressions for C_T are given in column two. The CA-STAR networks differ from the other networks in that the MAC of ordinary stations has lower computational complexity than that of the CA station. Thus we represent the time complexity of the CA-STAR networks by the complexity of their protocols for

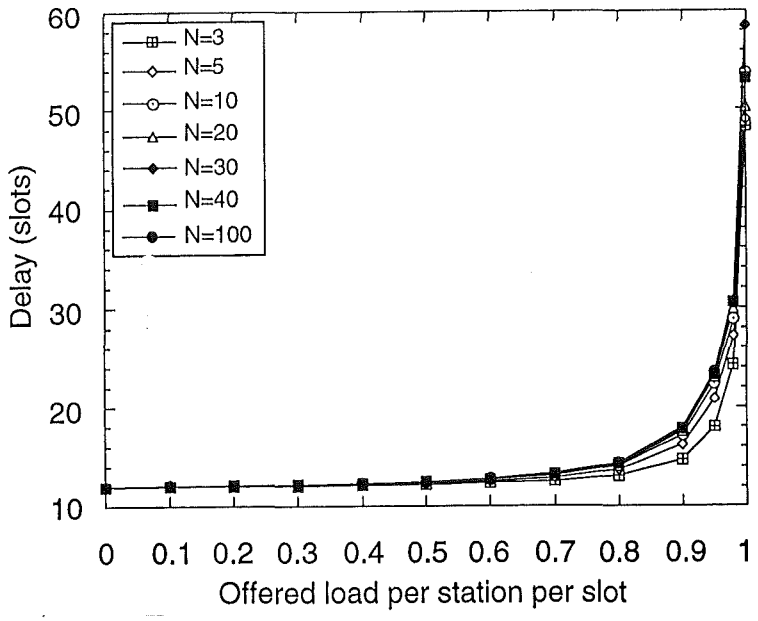


Figure 7.10: Mean packet delay of optCA-FPCF/R* versus load, for varying number of stations. B=25, a=5. Relative precision $\leq$5%.

optCA. One can find that the optCA-FPCF/B and optCA-FPCF/R protocols have lower order C_T than the protocol executed by stations in DAS (O(N^4)) and HTDM (O(N^4)) networks [CHIP93], and CF-WDMA (O(N^2)) [CHEN91] networks. optCA-FPCF/B (/R) has linear, i.e. O(N) complexity; the same as DT-WDMA [CHEN90]. Recall that optCA-STAR's performance remains near optimal for a fixed B, as network size increases. Thus we can treat B in the expressions for the computational complexity of optCA-FPCF/B and optCA-FPCF/R networks as a constant, even as N is increasing.

The C_T of CF-WDMA, DAS, HTDM, DT-WDMA, sCA/B, sCA/R, and optCA-MRS* networks were derived in Appendix F.

Almost all operations of the FPCF* based optCA-STAR protocols are matrix operations with potential parallelism that can be exploited by execution on a simple vector or multiprocessor system. The execution time of the FPCF based protocols can thus be further reduced. Also, many of the comparisons can be performed in parallel using associative testing. Since in optCA-STAR networks only one station - the optCA - is tasked with destination conflict resolution, vector or multiprocessing hardware supplied to just one station can speedup the protocol (electronic) execution time of the entire network. Economically, optCA-STAR is best positioned to benefit from the use of SIMD or MIMD hardware.

	optCA-FPCF/B	optCA-FPCF/R	
q	Prob(packet lost | q)	Prob(packet reflected | q)	R_m
1.00	0.01585 (0.01575, 0.01596)	0.1310 (0.1299, 0.1321)	8.27
0.99	0.01031 (0.01023, 0.01039)	0.03124 (0.03094, 0.03154)	3.03
0.98	0.006092 (0.006031, 0.006152)	0.01964 (0.01950, 0.01978)	3.22
0.97	0.003285 (0.003254, 0.003315)	0.01164 (0.01155, 0.01174)	3.54
0.96	0.001647 (0.001632, 0.001661)	0.006072 (0.006032, 0.006111)	3.68

Table 7.1: Multiplier effect of reflection, measured by the ratio of the probability of packet loss (optCA-FPCF/B) to the probability that a packet will be received by optCA for reflection (optCA-FPCF/R), as a function of q. An optCA-STAR network with N=10, B=40, and a=5 was assumed.

7.4 All Optical optCA-STAR Network using Wavelength Converters

An interesting property of optCA-FPCF/B and optCA-FPCF/R is that buffer module Q_i of optCA is used as a packet-carrying pipe with constant emptying rate. Physically, rescued packets are stored in buffer locations determined by FPCF*, and the packets in the E-th location of each buffer are transmitted during the same time-slot, with E rotated deterministically. But logically, it is correct to think of a buffer module as a pipe with B sections. Each section is either occupied by a packet, or is empty. Occupants of all sections flow towards the output at a *constant* rate. The rate of flow equals an advance of one section per time-slot. During a time-slot, the packet in the last section (output end) of the pipe is emptied (transmitted) on λ_{N+i}.

This property, combined with the fact that the channels at optCA are still space division multiplexed (at most one packet would arrive on the input fiber to a buffer module, and the arriving packets are on one wavelength), can be exploited to allow a simple implementation of optical buffer modules, provided that optical wavelength converters are available.

TABLE 7.4 ELECTRONIC PROCESSING COMPLEXITY OF VARIOUS WDM STAR NETWORKS

Network	Computational Complexity	
	C_T	C_N
Request-schedule-then-transmit (CF-WDMA)	$20N^2 + 3N$	$20N^3 + 3N^2$
Detect-and-retransmit (DT-WDMA)	$22N + 15$	$22N^2 + 15N$
Request-schedule-then-transmit (DAS)	$\sum_{i=1}^{N}\sum_{j=1}^{N}(N-i+2)(N-j+2)(N+i-1)$	$N\sum_{i=1}^{N}\sum_{j=1}^{N}(N-i+2)(N-j+2)(N+i-1)$
Request-schedule-then-transmit (HTDM)	$\frac{M}{N+M}\sum_{j=1}^{N}\sum_{k=1}^{N}(N-j+2)(N-k+2)(N+j-1)$	$\frac{NM}{N+M}\sum_{j=1}^{N}\sum_{k=1}^{N}(N-j+2)(N-k+2)(N+j-1)$
sCA-Star	$8N^2 + 18N + 2$	$12N^2 + 21N + 2$
optCA-Star (optCA-MRS*)	$23N^2 + 25N$	$23N^2 + 29N$
optCA-Star (optCA-FPCF/B)	$3N + 2B(N-1) + 3$	$7N + 2B(N-1) + 3$
optCA-Star (optCA-FPCF/R)	$6N + 2BN + 2$	$11N + 2BN + 2$

Table 7.2: Comparison of the worst case computational complexity of WDM star network protocols based on the request-schedule-then-transmit, detect-and-retransmit, and en-route-conflict-resolution (central placement) principles for destination conflict resolution.

7.4.1 Optical Buffer Modules

Construction

The configuration of an optical buffer module with a capacity for storing B packets is depicted in Fig. 7.11. It comprises a B stage pipe. Each stage is made of a delay loop (loop of fiber), and an ON-OFF switch. Denote stage j of buffer module Q_i by $Q_{i,j}$. Let $Q_{i,1}$ be the last stage in the pipe (i.e. the one at the output end), $Q_{i,2}$ be the second last stage, and so on. Denote the switch of $Q_{i,j}$ by $X_{i,j}$. At the output of the pipe lies a wavelength converter.

Each stage has 2 inputs and one output. The output of $Q_{i,j}$ is connected to one of the inputs of $Q_{i,j-1}$ (the next stage down the pipe). The exception is stage $Q_{i,1}$ (the last stage). Signals propagating from the output of this end

stage enters a wavelength converter Ω_i. Originally all signals in $Q_{i,1}$ are on λ_i. The Ω_i converts signals on λ_i to λ_{N+i}. The output signals from Ω_i are then combined with signals from the input using a 2×1 multiplexer, and the merged signal enters the i-th port of the star coupler. The other input of $Q_{i,j}$, j=2, 3, ... , B, is connected to the output of $X_{i,j}$.

The delay loop of each stage has a length set such that the propagation delay through the loop of fiber equals one time-slot. The input to Q_i is broadcasted[7] to all $X_{i,j}$, j=1,2, ..., B, and to one of the inputs of the 2×1 multiplexer. If $X_{i,j}$=ON, then the signal at its input is allowed to propagate to the input of $Q_{i,j}$. Otherwise it is absorbed.

Choosing a Procedure for Resolving Conflicts using an optCA with Optical Buffer Modules

The design of the optical buffers of optCA is influenced by the functionality that they should provide, which in turn depends on the choice of algorithm for destination conflict resolution at optCA. Previous algorithms are executed during one time slot, during which such an algorithm examines the inlet queues trying to maximise the number of packets which can be transmitted in a conflict-free way during the next time slot. In contrast, the FPCF algorithm considered in Chapter 6 achieves high throughput, low computational and logical buffer organisation complexity by planning conflict-free transmissions of all buffered packets for up to B-1 time slots into the future. The results suggests the choice of FPCF. However the main reason why FPCF was chosen is that the concept of *forward planning naturally facilitates a simple optical implementation* of optCA's buffers. Under FPCF, a new packet arriving during the current time slot has its transmission scheduled during one of the future B time slots, or is rejected. *Inverting our viewpoint, we can see that scheduling a packet for transmission (from optCA) during the j-th future time-slot is equivalent to scheduling its delay at optCA by j time slots (j=1, 2, ... , B).* This permits a simple optical implementation of the buffers, as the functionality required of the buffers are much reduced. For example, in other algorithms, the duration which a packet has to be buffered is unbounded and unknown until just before its transmission. With FPCF, each packet requires buffering for at most B time-slots and the duration of stay is know prior to its entry into the buffers of CA. Consequently, the storage functions needed for operations under FPCF can be served by a simple series of delay lines, as

[7] A B-way optical switch can be used in place of the B ON-OFF switches. Its use would avoid the power loss from splitting the input signal power into B+1 parts.

described in the previous section.

Mapping the optCA MAC Protocols onto an optCA using Optical Buffering

Under optCA-FPCF/B and optCA-FPCF/R, a packet placed in the j-th location of Q_i would be transmitted during the D(j,E)-th future time-slot, where

$$\begin{aligned} D(j,E) &= B+(E-j) \quad if\ (E-j)<0 \\ & \quad\ E-j \quad o.w. \end{aligned} \tag{7.6}$$

When optical buffers are used in place of electronic memory, an optical buffer module Q_i is accessed as follows. At the beginning of every time-slot t, all $X_{i,j}$, j=1,2, ..., B are in the OFF position by default. Consequently, the input signal to Q_i (from S_i) will enter the star coupler without delay by default. If a packet needs to be rescued, then $1/(B+1)$ -th fraction of the input power is directed to the input of one of the buffer stages as follows. Suppose that FPCF* decided that during t, a packet from S_i needs to be rescued and placed in the j-th location of Q_i. Define L=D(j,E). Then the following step is actioned to rescue the packet for transmission during the scheduled future time-slot :

- set $X_{i,L}$ to ON

No reception is necessary. No transmission is necessary. No electronic memory is needed for storing rescued packets. Note that if a packet would be successful (does not require rescuing), then L=0, so $X_{i,j}$=OFF, and no part of the signal power of the packet will enter the delay stages. If $L > 0$, the packet needs rescuing and buffering until t+L. By setting $X_{i,L}$=ON, the packet would propagate through L stages[8]. In effect, this schedules the packet for propagating from optCA to its destination during the L-th future time-slot, and on wavelength λ_{N+i}. In the meantime, the packet remains in the optical domain.

[8] $1/(B+1)$ fraction of the input signal power is always merged with the output from the wavelength converter, and the combined signal always enters the star coupler. Thus the signal of the packet will be available for reception by its intended destination during its original time of arrival. If the packet was also rescued then the destination could not receive the packet during its original time of arrival. Instead, the fraction of its signal which propagated through the buffer stage(s) will arrive when its destination is free to receive it.

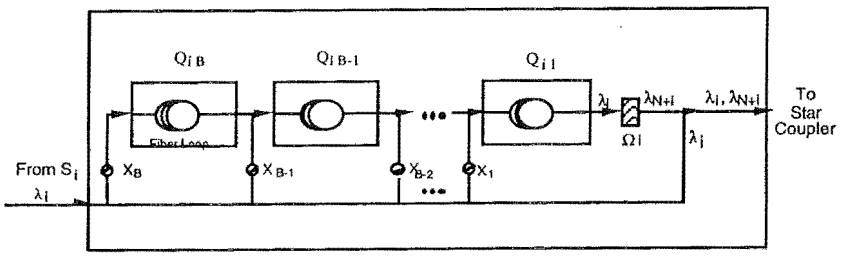


Figure 7.11: The configuration of an optical buffer module using wavelength converters.

Notwithstanding, it should be pointed out that the implementation of optCA using electronic memory assumed heretofore is already optimal in terms of performance. If a packet needs rescuing by optCA, it would not have been received by its destination during its original time-slot of arrival. Hence, rescued packets need to wait at least one time-slot before their destinations would be free. The total delay caused by E/O conversion at optCA is one time-slot, so the delay is desirable. The use of electronic memory by optCA to store rescued packets therefore does not degrade the performance of optCA-STAR networks. Nor does the use of electronic buffering limit the scalability of optCA-STAR. Advances in technology that enable the interface of ordinary stations to increase their transmission rate would enable optCA to match the improved rate, since the technology and the complexity of the busing structure of the interface of optCA and ordinary stations are identical. The performance of all optical optCA-STAR networks is identical to those which use electronic buffering, since the optical buffers provide the same logical functionality.

Optical buffering of packets is thus considered in this section only to provide an alternative implementation of optCA-STAR, not to improve optCA-STAR performance. The all optical optCA-STAR alternative allows the engineer to choose an implementation based on costs and reliability considerations.

7.5 Conclusions

In WDM networks the computational complexity of the MAC protocol and the complexity of the logical buffer organisation can become a critical concern. The execution time of the MAC protocol is typically restricted to one time-slot. As the number of stations grows, the worst case computational complexity of the MAC procedure executed by stations may increase, but the time required for executing it remains unchanged. Thus a protocol with high complexity growth may itself develop into a bottleneck. Similarly, logical buffer organisation requires resources (for its implementation) and execution time (for maintaining it). Previous networks require the packets that are waiting in transmit buffers of a station to be organised into one or more logical structures. A logical buffer structure defines relationships between the packets stored in the physical buffer. Logical buffer structures are used to support operations required by the MAC protocol of stations. For instance, a common logical organisation of packets in the transmit buffer of a station is to arrange them into N-1 FIFO queues, one queue for packets destined for a given destination station. Logical buffer organisation consumes both resources (pointers) and time (updating pointers). Moreover, data paths must be provided between the memory module where the pointers are stored, and the processing unit, and the buffer memory itself.

This chapter evaluated two optCA-STAR MAC protocols based on the Forward Planning Conflict-Free (FPCF) traffic assignment algorithm. FPCF is known to have low computational complexity, and requires extremely simple buffer organisation. The protocols, called optCA-FPCF/B and optCA-FPCF/R, were analysed considering their computational complexities, buffer organisation complexity, and throughput and mean packet delay performance.

The worst case time computational complexity of the optCA-FPCF/B, and optCA-FPCF/R protocols were of the order O(N), compared with O(N^4) for DAS and HTDM (request-schedule-then-transmit based networks), O(N^2) for CF-WDMA (request-schedule-then-transmit) and optCA-MRS* (en route conflict resolution), and O(N) for DT-WDMA (detect-and-retransmit). optCA-FPCF/B and optCA-FPCF/R have significantly lower computational complexity since only newly rescued packets by optCA need to be scheduled. Packets left in optCA from previous time slots are already scheduled for transmission during one of the future time slots. optCA-FPCF/B and optCA-FPCF/R seem especially well positioned to exploit SIMD, MIMD, or associative testing hardware for further reductions in the time needed for their computation. Due to complexity inversion, such hardware would be needed at just one station

(the optCA is the only station tasked with destination conflict resolution) instead of all network stations.

FPCF uses the knowledge of future transmissions to streamline logical buffer organisation. Under optCA-FPCF/B and optCA-FPCF/R, all packets in optCA that are scheduled for transmission during a time slot are identified by an index (E) which rotates *deterministically* after each slot. There is no need to search the buffer modules for the locations of packets that are scheduled for transmission. Thus, no queues nor lists of packets need to be created nor maintained. No logical relationships between packets in a buffer needs to be recorded. Planning the future transmission of a packet becomes equivalent to determining the physical address for storing a packet. Logical buffer organisation is therefore completely subsumed by the Forwarding Planning of packet transmissions.

optCA-FPCF/B and optCA-FPCF/R also achieved near optimal throughput and mean packet delay, provided that the offered load was not too high (below 90%). Note that according to optCA-FPCF/B, and optCA-FPCF/R, optCA transmits packets that it had rescued after a delay of at most B-1 time slots. Provided that the offered load is below 95% (probability of Reflection is low), this should yield a reduction of the delay variance with respect to networks using the SDR, MRS or RS algorithms for conflict-free transmission scheduling.

Technologies needed for optCA's construction are identical to that required for the network interface of ordinary optCA-STAR stations. Thus any technological advancements that improve the data rate of ordinary stations should also enable optCA to match the improved rate. In optCA-FPCF/B (/R) networks each buffer module of optCA is used as a packet-carrying pipe with constant emptying rate. This property allows a simple implementation of optical buffer modules resulting in an all-optical network based on en route conflict resolution. However, it must be noted that the electronic buffer modules are already ideal in terms of their support of en route conflict resolution as implemented in the optCA-FPCF/B (/R) networks. Buffering "otherwise lost" packets optically instead of electronically does not improve the performance nor scalability of the networks. Also, each optical buffer module requires the use of one wavelength converter.

Additional advantages of the optCA-STAR architecture are high modularity and improved upgradability. For example optCA can be upgraded to serve a growing network by adding one buffer module per new station. A fault in an optCA module only affects the corresponding station, and it is easier to increase the memory capacity of buffers by adding components at just one site

- the optCA, instead of increasing the memory capacity of all network nodes.

Chapter 8

FPCF optCA-STAR Networks with Reduced Number of Channels

All the preceding CA-STAR networks assumed the availability of $2N$ channels in a network with N stations. Also, the buffer design of optCA introduced previously is suitable for optical implementation only if wavelength converters are available. This chapter considers a simple extension to the optCA-STAR architecture, which allows the use of only N channels thereby halving the network's bandwidth requirement, and which naturally fits the use of optical buffers for storing "otherwise lost" packets, thereby yielding an all-optical CA-STAR networks that does not depend on the availability of wavelength converters. The newer architecture is briefly discussed next, followed by modifications to the optCA-FPCF protocols. Following the convention set out in Chapter 1, the architecture will be named rcCA-STAR ("rc" denoting reduced channels), and the protocols[1] will be named rcCA-FPCF/B and rcCA-FPCF/R. Also following previous convention, we shall refer to an rcCA-STAR network that is operating according to the rcCA-FPCF/B (/R) protocol as an rcCA-FPCF/B (/R) network.

[1] A list of all protocols considered in this thesis is already contained in Chapter 1, 9, and 10. It is neither necessary nor appropriate to list them here again.

8.1 optCA-STAR Architecture with Reduced Channels

The architecture of optCA-STAR networks using a reduced number of channels (rcCA-STAR) is identical to that of optCA-STAR, see section 5.1, except for two changes. First, the number of data channels required is reduced to from $2N$ to N. Secondly, an optical ON/OFF switch is added to the buffer modules of the central arbiter station.

Accordingly, an rcCA-Star network has N stations S_i, for i =1, 2, ... , N; N+1 WDM channels, λ_c and λ_i (for i =1, 2, ... , N); a passive star coupler, and a central arbiter station (rcCA), refer to Fig. 8.1. The rcCA is located at the entrance to the star coupler.

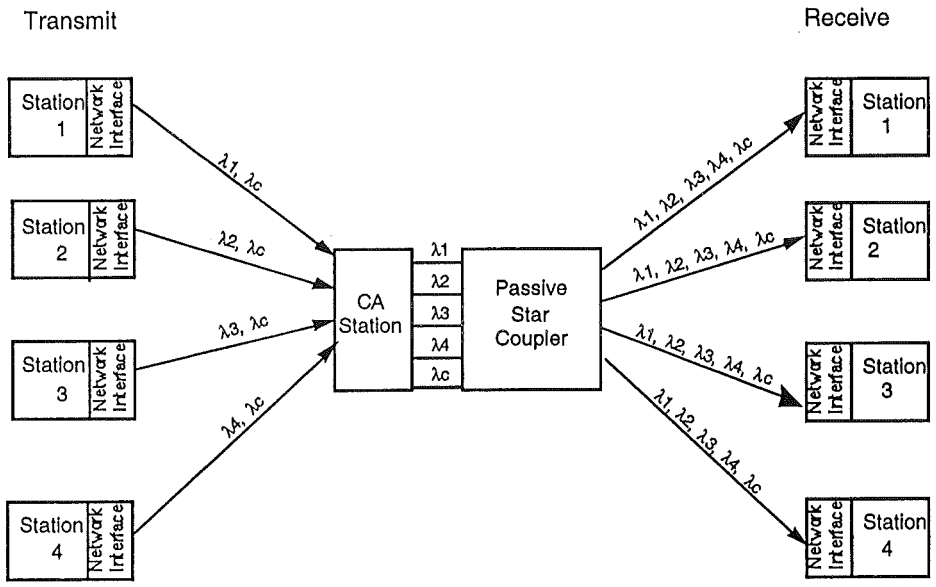


Figure 8.1: Logical architecture of an rcCA-STAR network with N=4 stations

S_i is connected directly to input port I_i of rcCA by an optical fibre, called S_i's *outgoing fiber*. This carries signals from S_i to rcCA. rcCA has N+1 outputs, one for carrying data from each station, plus one for sending control signals. These are connected to the input ports of an $(N+1)\times(N+1)$ star coupler. The star coupler combines optical signals from its N +1 input ports and broadcasts them to its N +1 output ports, as explained in section 3.2.1. A fibre runs from each output port O_i of the star coupler to each S_i. It is called S_i's *incoming fiber*. It carries the combined signals from rcCA to S_i.

8.1.1 Channel Structure

Stations and rcCA are synchronised, and channels are time slotted. The duration of a slot equals the transmission time of one (fixed length) packet, plus the tuning period [CHEN90], [HUMB93], [CHEN91], [CHLA91], [CHIP93], [CHEN92].

Slots on channels λ_1, λ_2 , ..., λ_N are called *data slots*. Each data slot can carry one data packet, see Fig. 8.2.

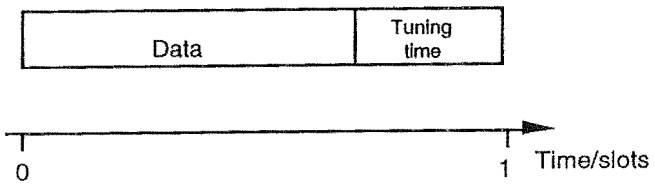


Figure 8.2: Format of a data slot.

λ_c is the common control channel. Slots on λ_c are called *control slots*. Each control slot is subdivided into $2N$ mini-slots, see Fig. 8.3. Mini-slot i (i =1, ..., N) can carry the address of one station. Mini-slot j (j =N+1, ..., $2N$) can carry the index of a data channel. Thus the channel structure of an rcCA-STAR network is as described in Chapter 5 and 7 for optCA-STAR networks, except that all mini-slots are $\log_2(N)$ bits wide (in optCA-STAR networks, mini-slots N+1 to $2N$ are $\log_2(2N)$ bits wide, i.e. one bit wider than mini-slots 1 to N)

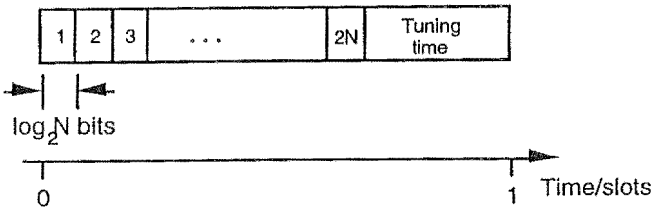


Figure 8.3: Format of a control slot.

Each station accesses its incoming and outgoing fibers through a network interface.

8.1.2 Station Network Interface

The network interface of stations in an rcCA-STAR network is identical to that of optCA-STARs, see section 5.1.2. Since the number of data channels used in rcCA is half that of optCA, the tuning range of the data receiver of stations in rcCA-STAR can be reduced by half.

8.1.3 The Architecture of the Central Arbiter Station of rcCA-STAR

The central arbiter station of rcCA-STAR (rcCA) is made of N buffer modules, plus one fixed tuned transmitter and receiver for accessing the common control channel, see Fig. 8.4. Input I_i to rcCA carries data on λ_i and control signals on λ_c. Signals on λ_c from all inputs are coupled to the input of the control receiver. Signals on λ_i (of I_i) enter the ith buffer module.

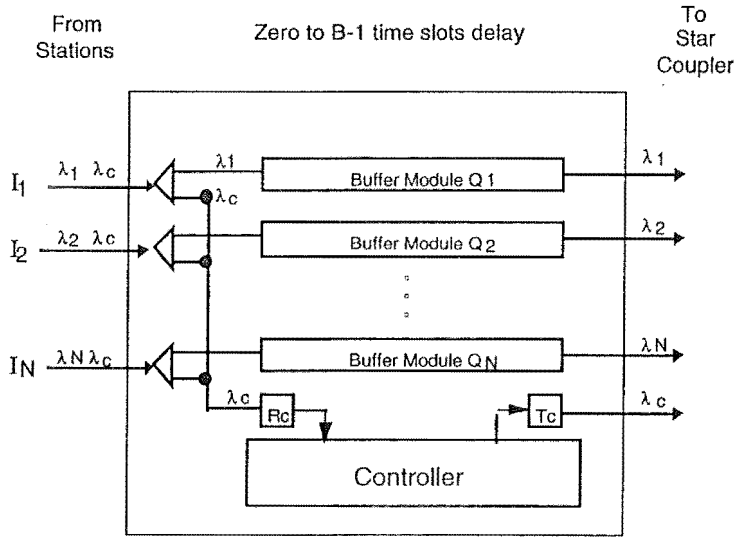


Figure 8.4: The Central Arbiter station for a rcCA-STAR network with N stations. R=fixed tuned receiver, T=fixed tuned transmitter.

Each buffer module Q_i (i=1,2, ..., N) is made of one receiver R_i, one memory module M_i, one fixed tuned transmitter T_i, and one optical ON/OFF switch X_i, see Fig. 8.5.

The optical input into Q_i is split using a directional coupler, with $N/(N+1)$ fraction of the input signal's power directed to the ON/OFF switch X_i. If X_i

is ON, that signal will enter the ith port of the star coupler. Then packets from S_i remain in the optical domain, bypassing rcCA. If X_i is OFF, the bypass would be blocked (signal absorbed).

The remaining $1/(N+1)$ of the input power always enters R_i. R_i is for receiving packets arriving on λ_i, i.e. S_i's data channel, whenever X_i is OFF. All packets received by R_i are stored in M_i. These packets can be transmitted from M_i on data channel λ_i, using T_i.

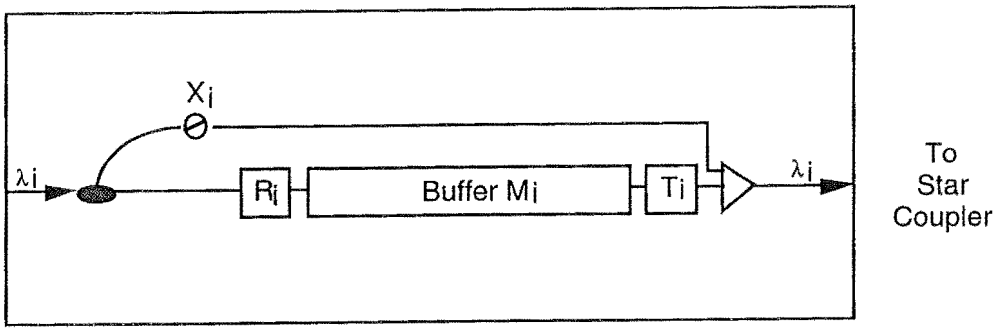


Figure 8.5: Block diagram of buffer module Q_i at rcCA

8.1.4 The Design Rationale of rcCA

Clearly, rcCA-STAR is an extension of optCA-STAR. In optCA, $N/(N+1)$ fraction of the input power to Q_i (carrying data from S_i) is always directed to the ith input port of the star coupler. In contrast, with rcCA, $N/(N+1)$ fraction of the input power to Q_i enters the input of an ON/OFF optical switch, X_i. If X_i is ON, that fraction of the signal's power enters the ith input port of the star coupler, as in optCA. However, if X_i is OFF, that fraction of the signal power is absorbed. In both CAs, the remaining $(1/(N+1))$ fraction of the signal power entering Q_i is always directed to R_i.

rcCA rescues packets that would otherwise be lost due to destination conflicts, re-scheduling their arrival times so that they reach their destinations when their destinations are free to receive them A packet from S_i bypasses rcCA (i.e. X_i is ON), if it would be successful. Otherwise, X_i is OFF and the packet is rescued (received) by R_i into buffer M_i, and waits there until it can be transmitted to its destination without a conflict.

Several devices can serve as an optical ON/OFF switch, such as the magneto-optic effect light modulator [ROSS82] currently in the market place, or the digital optical switch [CAVA91], see section 3.2.4.

8.2 rcCA-STAR Protocols

The general MAC principle for rcCA-STARs is similar to that for optCA-STAR networks. A ready station firstly announces its intention to transmit a packet on the control channel, then transmits the packet on its data channel after one time slot. Destination conflicts occur if two or more packets are destined for the same station during the same time slot. rcCA detects destination conflicts, and reschedules the arrival times of "otherwise lost" packets to the soonest time-slot when their destinations are free to receive them. Ordinary stations listen to the control channel to determine which data channel to receive from.

However, only N data channels are available in rcCA-STAR networks. Station S_i shares λ_i with data transmitted from the ith buffer of rcCA. rcCA may thus wish to transmit a packet in a slot on λ_i, when that slot contains a packet from S_i. In this case, we assume that rcCA's transmission should succeed. rcCA sets X_i to OFF during the slot, thus preventing S_i's packet from entering the star coupler. rcCA then transmits its packet during the slot, and rescues the packet (from S_i) it displaced in the usual way.

As was done for sCA-STAR and optCA-STAR networks, two protocols will be considered for the operation of rcCA-STAR networks. The first, named rcCA-FPCF/B, guarantees that the delay of a packet is less than or equal to $2a+B$ time slots, and that packets from a source station that are transmitted to a specific destination will be delivered in the order in which they were transmitted. Under rcCA-FPCF/B, packets may be lost, although the probability of packet loss can be kept below a specified value through the use of a connection acceptance function, as discussed in section 1.4.2. The second protocol is named rcCA-FPCF/R. Under rcCA-FPCF/R packet loss is prevented using the Reflection mechanism.

8.3 The rcCA-FPCF/B Protocol

As in previous networks which are based on en route conflict resolution, the MAC protocol for ordinary stations can be considerably simplified. The rcCA-FPCF/B protocol for rcCA-STAR networks therefore comprises a MAC procedure for ordinary stations, and one for rcCA.

8.3.1 The MAC Protocol of Ordinary Stations

The MAC protocol for ordinary network stations is identical to the one used in optCA-MRS, see section 5.2.2 on page 115.

8.3.2 MAC Protocol for the rcCA Station according to rcCA-FPCF/B

The MAC protocol for the rcCA station is as defined for optCA-FPCF/B (section 7.2 on page 157), with the addition of two statements. One is for setting X_is. The second is to include any displaced packets for buffering by rcCA. These two changes will be highlighted by underlining in the following definition. The resulting protocol, with these two changes is as follows.

Let each of the N buffer modules of rcCA be sized to store B packets, $B \geq 2$. Think of the N rcCA buffers as a matrix $C=[c_{ij}]_{N \times B}$ of packet size memory locations (Fig. 8.4) where (i) there can be at most one "write" (receive) and one "read" (transmit) operation per row, and (ii) a read and a write operation on the same row must be on distinct locations.

Definition of the rcCA MAC Protocol

Let rcCA maintain the following variables

- Planned Reception Matrix, $P = [p_i]_{N \times 1}$. At the beginning of t, $p_i = k$ if during t, the packet transmitted by S_i should be received by rcCA into location k of the ith buffer module; $p_i = 0$ o.w..
- Next Reception Matrix, $P^+ = [p_i^+]_{N \times 1}$. At the beginning of t, $p_i^+ = k$ if during $t+1$ rcCA should receive the packet from S_i into the location k ; $p_i^+ = 0$ o.w..
- Mini-slot Transmission Matrix, $M = [m_i]_{N \times 1}$. m_i is the channel number which would be transmitted by rcCA on the $(N+i)$-th mini-slot during the current time slot.
- Enabled Column Counter, E. E is initialised to B during network initialisation, and decremented by one after each time slot, and if $E==0$ then E is reset to B .

Procedure rcCA Transmission (Executed during each slot)
CoBegin
transmit m_i on the $N+i$th mini-slot of the current control slot, **for** i=1, ..., N ;
forall packet in c_{iE}, $i = 1, ..., N$ **doparallel**
transmit packet in c_{iE} on λ_i ;
CoEnd
$E = E$ - 1 ; **if** $(E == 0)$ **then** $E = B$;
forall X_i, $i = 1, ..., N$ **doparallel**
if (c_{iE} >0) **then** X_i=OFF; **else** X_i=ON ;

Procedure rcCA Reception (Executed during each slot)
CoBegin
forall $i = 1, ..., N$ **doparallel**
{ **if** $(p_i \neq 0)$ **then** receive the packet transmitted by S_i into c_{i,p_i},
receive mini-slots 1 to N ; $P = P^+$; update M and P^+ using the FPCF$^*_{rc}$ algorithm ;
CoEnd

8.3.3 FPCF$^*_{rc}$ Algorithm

The FPCF$^*_{rc}$ algorithm is used for planning the reception of "otherwise lost" packets (including those displaced by rcCA) into rcCA, and for scheduling received packets for conflict-free transmission to their destinations. FPCF$^*_{rc}$ is almost identical to FPCF*. Only the rule for determining which packets need rescuing by rcCA is extended to included packets that would otherwise be lost, because rcCA planned to transmit on the channels that they occupied. Specifically, if λ_i contained a packet from S_i, and rcCA had planned to transmit a packet from Q_i during the same time slot, then that data slot (on λ_i) would be emptied by setting X_i to OFF, as defined in procedure rcCA Transmission. rcCA can therefore use the emptied slot, and rescue the packet from S_i. The definition of FPCF$^*_{rc}$ is as follows (the modified statement is underlined).

The additional variables used by FPCF$^*_{rc}$ are:

- Future C Status Matrix, $F = [f_{ij}]_{N \times B}$. At the end of the FPCF-processing-time-period of t (see Fig. 8.6), f_{ij} = destination address of the packet that will be in c_{ij} during t+2. If c_{ij} will be empty during t+2 then $f_{ij} = 0$.

- Destination Allocation Matrix, $\Delta = [\Delta_{ij}]_{N \times B}$. During t , $\Delta_{ij} = 1$ if one of f_{1j}, f_{2j}, ..., f_{Nj} equals i. $\Delta_{ij} = 0$ o.w..
- FaVoured Station (V) Counter. Initialised to 1 during network startup.
- Future E counter, E^+. E^+ is the value of E two time slots from now. E^+ is initialised to B-2.

Let $H = [h_1, h_2, ..., h_N]$ be the N addresses in the first N mini-slots of the current control slot.

procedure FPCF$^*_{rc}$(input H) // Compute new P^+ and M.
BEGIN
$V = V + 1$; **if** $(V > N)$ **then** $V = 1$;
$E^+ = E^+$ -1; **if** $(E^+ == 0)$ **then** $E^+ = B$;
for each S_i $i = V, V+1, ..., N, 1, 2, ..., V$ -1 **do**
{ **if** $h_i \neq 0$ **then** // S_i plans to transmit a packet
if $(\Delta_{h_i,E^+} == 1)$ **or** $(c_{i,E^+} > 0)$ **then** // packet will be displaced or its destinati
// already allocated. S_i 's packet will not be received by its intended destina
{ find the first j s.t. ($f_{ij} == 0$ **AND** $\Delta_{h_i,j} == 0$), searching in the order
given by $j = E^+$ -1, ..., 1, B, B -1, ..., E^+ +1 ;
if found, **then** $f_{ij} = h_i$; $\Delta_{h_i,j} = 1$; $p_i^+ = j$;
else $p_i^+ = 0$; // can't find a conflict free location to buffer packet
} // thus rcCA cannot receive it. Packet will be lost (!)
else // S_i 's packet will be successfully received. No need to buffer by rcCA.
{ $p_i^+ = 0$; $\Delta_{h_i,E^+} = 1$; $m_{h_i} = i$; }
if ($f_{i,E^+} > 0$) **then** $m_{f_{i,E^+}} = i$;
} // endfor
$[f_{1,E^+}, f_{2,E^+}, ..., f_{N,E^+}] = [0, 0, ..., 0]$; $[\Delta_{1,E^+}, \Delta_{2,E^+}, ..., \Delta_{N,E^+}] = [0, 0, ..., 0]$;
END;

The role of counter V is to rotate the order in which stations are processed during each slot. The sooner a station is processed, the greater the probability that its packet could be buffered by rcCA, if its packet needed buffering. Rotating the processing order ensures that stations are fairly treated.

8.3.4 Providing Delivery Guarantees

According to rcCA-FPCF/B a packet p_i transmitted by S_i that needs buffering at rcCA would be lost, if FPCF$^*_{rc}$ cannot find an empty-and-conflict-free location in the ith row of C. As usual, the rcCA-FPCF/B protocol can be

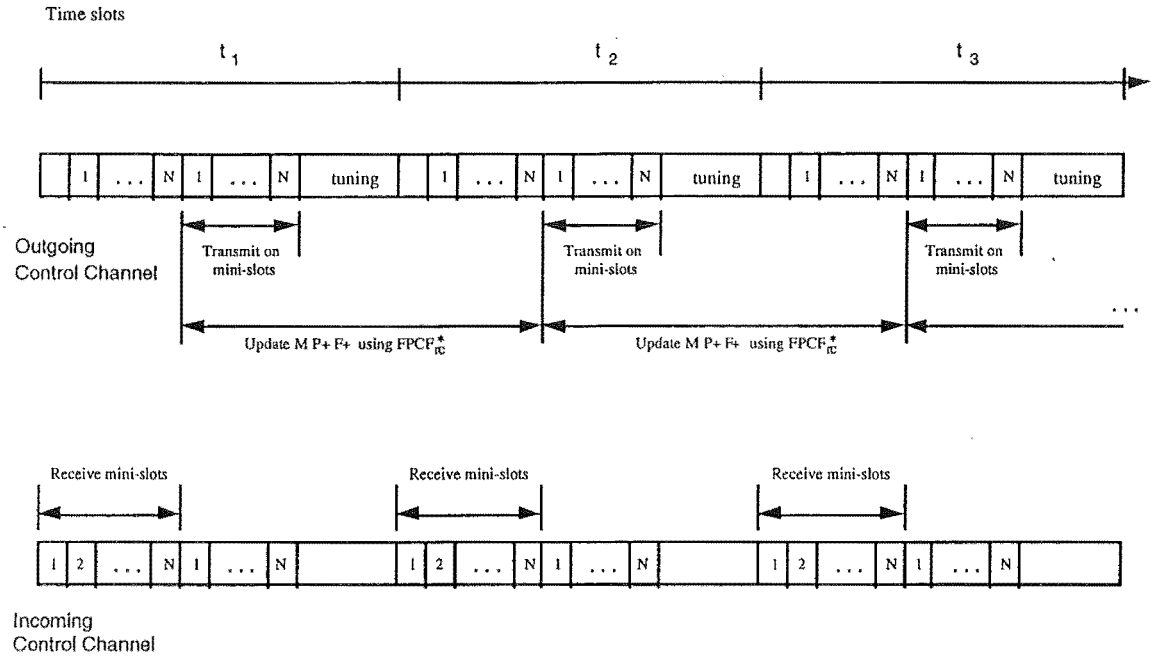


Figure 8.6: Timing of control channel operations and $FPCF^*_{rc}$ processing according to the rcCA-FPCF/B protocol.

extended to eliminate any loss of packets using Reflection (see section 7.2.4). The resulting protocol is named rcCA-FPCF/R.

8.4 Performance Evaluation

Results for the steady-state throughput and mean packet delay in rcCA-FPCF/B(/R) networks were obtained using the same method (section 4.2.7) and modelling assumptions (section 4.2.7).

8.4.1 Effect of Buffer Size on Throughput and Delay Characteristics

Results for rcCA-FPCF/B are plotted in Fig. 8.7 and Fig. 8.8, and results for rcCA-FPCF/R are plotted in Fig. 8.9 and Fig. 8.10, assuming N=10 stations, all a=5 slots from rcCA.

Results show that with channel sharing, it is still possible to obtain near optimal throughput (over 95% channel utilisation), and near minimum mean

packet delay, providing that the offered load was below 95% of channel capacity, with either rcCA-STAR protocol.

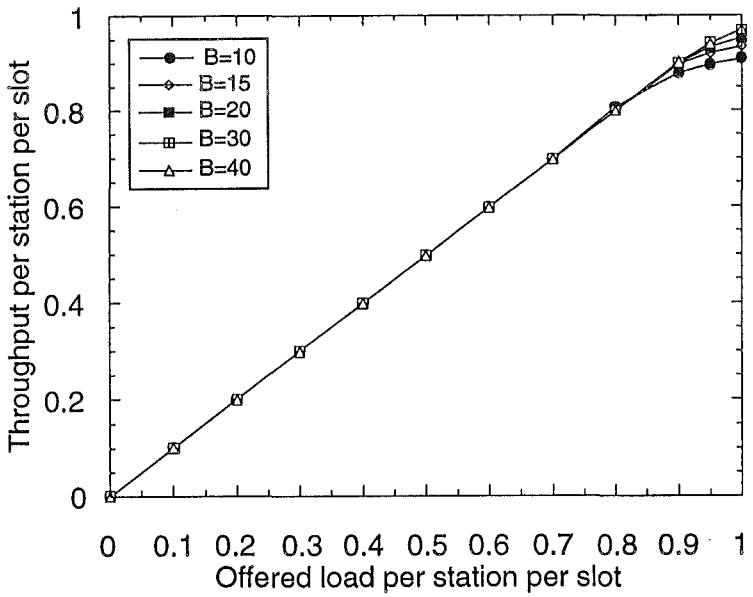


Figure 8.7: Throughput of rcCA-FPCF/B versus load, for varying buffer sizes. N=10, a=5. Relative precision $\leq$5%.

8.4.2 Impact of Increasing Network Size:

1) rcCA-FPCF/B: Fig. 8.11 shows that increasing network size from N =10 to 100 stations does not affect its efficiency. Increasing N from 3 to 5 stations increases average packet delay somewhat at medium traffic, see Fig. 8.12, but further increase has little effect. These results demonstrate that rcCA-FPCF/B remains a good solution as network size increases, *even when B remains constant.*

2) rcCA-FPCF/R: Figs. 8.13 and 8.14 reports the performance of rcCA-FPCF/R under the same range of q and N as that considered in the study of rcCA-FPCF/B above. One can see that rcCA-FPCF/R also enjoys near optimal performance irrespective of the network size when the offered load is between 0 and 95% of channel capacity, *even when B remains constant.* Above q =0.95, there is some degradation in throughput and delay, irrespective of N, as expected.

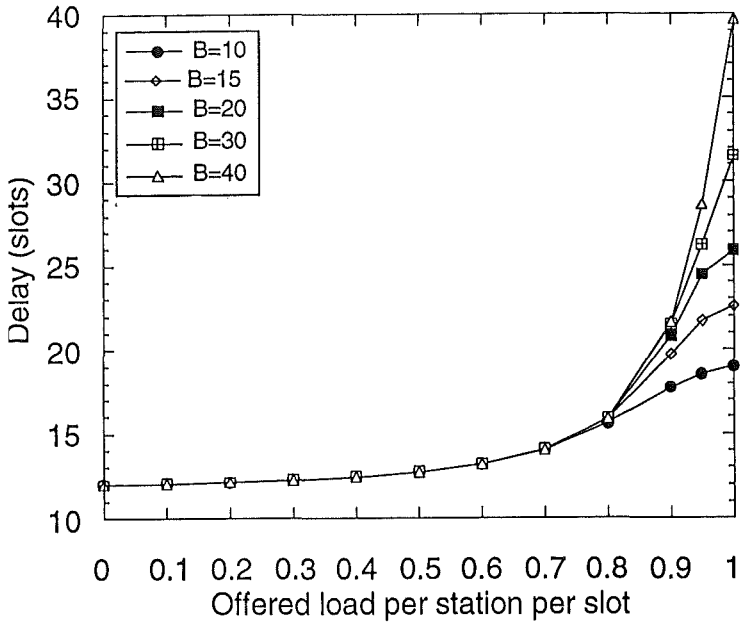


Figure 8.8: Mean packet delay of rcCA-FPCF/B versus load, for varying buffer sizes. N=10, a=5. Relative precision $\leq$5%.

8.4.3 Impact of Reflection on Performance

As was done for optCA-FPCF/R, the cost of employing Reflection is quantified by considering the *Reflection Multiplier factor*, $R_m\ (q\ )$, defined as

$$R_m\ (q\ ) = \text{Prob(packet reflected} \mid q\ ) \ /\ \text{Prob(packet lost} \mid q)$$

where Prob(packet lost | q) denotes the probability that a packet from a given station is lost during one time slot, when the network operates following rcCA-FPCF/B, at load q. Similarly, Prob(packet reflected | q) denotes the probability that a packet from a given station is reflected during a time slot, when the network operates following rcCA-FPCF/R, at load q .

Estimates of Prob(packet lost | q), Prob(packet reflected | q), and $R_m\ (q)$, for a rcCA-STAR network with N=10 stations, a=5, and B =40 are contained in Table 8.1. An examination of Table 8.1 suggests that Reflection does have a significant multiplier effect when the offered load is high. At the highest possible traffic level (q=1.0), $R_m(1.0) \approx$ 8.5.The interpretation is that when rcCA buffers one packet for Reflection (instead of loosing it), on average it creates the need to buffer 7.5 other packets for Reflection. Fortunately, the results show that the effect decreases to about 3 for $q \leq$0.99. This suggests that the overhead of Reflection is low except under maximum load. Moreover, the probability that Reflection occurs in rcCA-FPCF/R diminishes quickly, from 0.20 at maximum load, to 0.0060 at q=0.95. This explains the near

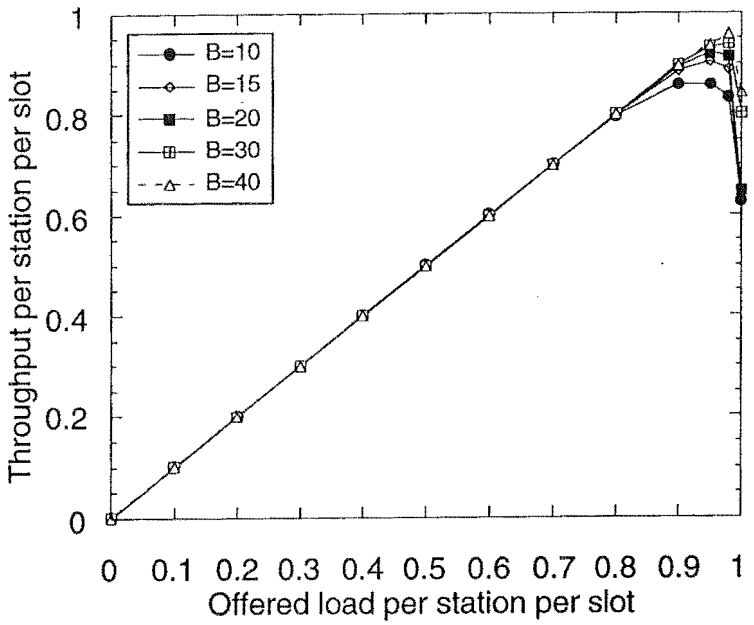


Figure 8.9: Throughput of rcCA-FPCF/R versus load, for varying buffer sizes. N=10, a=5. Relative precision $\leq$5%.

optimal performance of rcCA-FPCF/R provided the offered load is not too high.

8.4.4 Analysis of Computational Complexity

Let *network computational complexity* (C_N) and *time computational complexity* (C_T) be as defined in section 7.3.4.

Table 9.5 compares C_N and C_T electronic processing complexities of the rcCA-STAR networks with previous CA-STAR networks, and with that of networks using the request-schedule-then-transmit or the detect-and-retransmit-if-lost schemes. The comparison is based on a network with N stations, each with a transmit buffer of size B.

We represented the C_T of the rcCA-STAR networks by the complexity of the MAC protocol of the rcCA station executing the rcCA-FPCF/R protocol (more complex of the two presented in this chapter). One can find that the rcCA-FPCF/B and rcCA-FPCF/R protocols have lower order C_T than the protocol executed by stations in DAS (O(N4)) and HTDM (O(N4)) networks [CHIP93], and CF-WDMA (O(N^2)) [CHEN91] networks. rcCA-FPCF/B (/R) has linear, i.e. O(N) complexity; the same as DT-WDMA [CHEN90] and the optCA-FPCF/B and optCA-FPCF/R networks. Recall

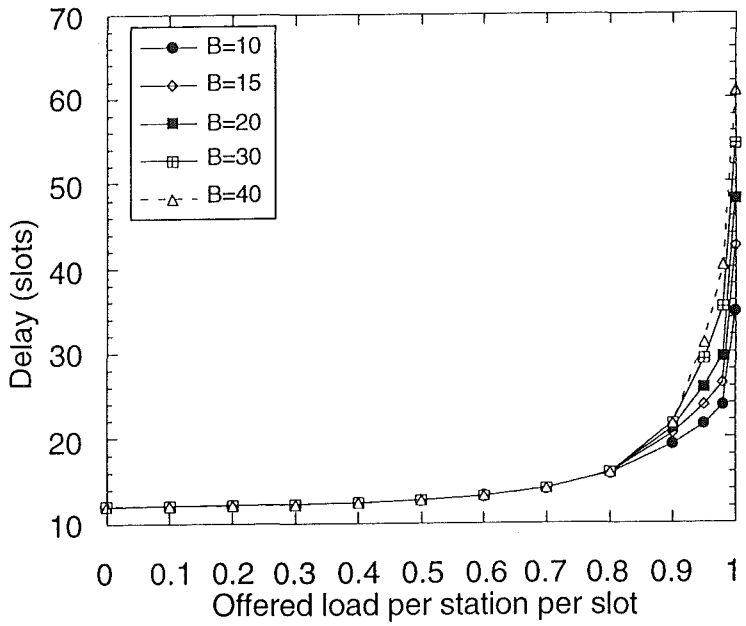


Figure 8.10: Mean packet delay of rcCA-FPCF/R versus load, for varying buffer sizes. N=10, a=5. Relative precision $\leq$5%.

that rcCA-STAR's performance remains near optimal for a fixed B, as network size increased. Thus we can treat B in rcCA-STAR's complexity expression as a constant, even as N is increased.

The differences in the time computational complexity between rcCA-FPCF/R and optCA-FPCF/R is due to the $2N$ additional comparisons performed when executing rcCA's MAC protocol. N of the operations are performed in rcCA's Transmit procedure (testing $c_{i,E}$ with zero to determine whether to set X_i ON or OFF). The remaining N comparison operations are executed in the FPCF$^*_{rc}$ procedure (testing the truth of the inequality $c_{i,E^+} > 0$).

8.5 Using Optical Buffering

rcCA-STAR has two further properties of interest. One is that once a packet is transmitted on a wavelength (channel), it will always be delivered on that wavelength. That is, a packet would never require wavelength conversion, even if it needs rescuing by rcCA.

The other property is that buffer module Q_i of rcCA is used as a pipe with constant emptying rate. Physically, rescued packets are stored in buffer locations determined by FPCF$^*_{rc}$, and the packets in the E-th location of each buffer are transmitted during a time-slot, with E rotated deterministically.

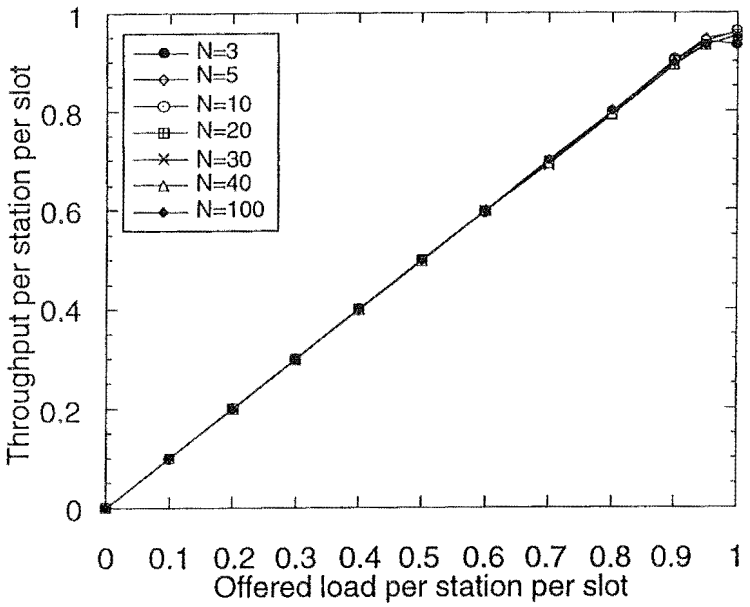


Figure 8.11: Mean packet delay of rcCA-FPCF* versus load, for varying number of stations. B=25, a=5. Relative precision $\leq$5%.

But logically, it is correct to think of a buffer module as a pipe with B sections. Each section may be occupied or empty. Occupants of all sections flows towards the output at a *constant* rate. The rate of flow equals an advance of one section per time-slot. During a time-slot, the packet in the last section (output end) of the pipe is emptied (transmitted) on λ_i.

These properties, combined with the fact that, at rcCA, channels are still SDM (at most one packet would arrive on the input fiber to a buffer module, and the packets arriving on a given fiber are on one wavelength), can be exploited to allow a simple implementation of optical buffer modules using well established technologies.

8.5.1 Optical Buffer Modules

Construction

The configuration of an optical buffer module with a capacity for storing B packets is depicted in Fig. 8.15. Its design is simpler than that of the optical buffer modules of an optCA-STAR network, since no wavelength converters are required. It comprises a B stage pipe. Each stage is made of a delay loop (loop of fiber), and an ON/OFF switch. Denote stage j of buffer module Q_i by $Q_{i,j}$. Let $Q_{i,1}$ be the last stage in the pipe (i.e. the one at the output end),

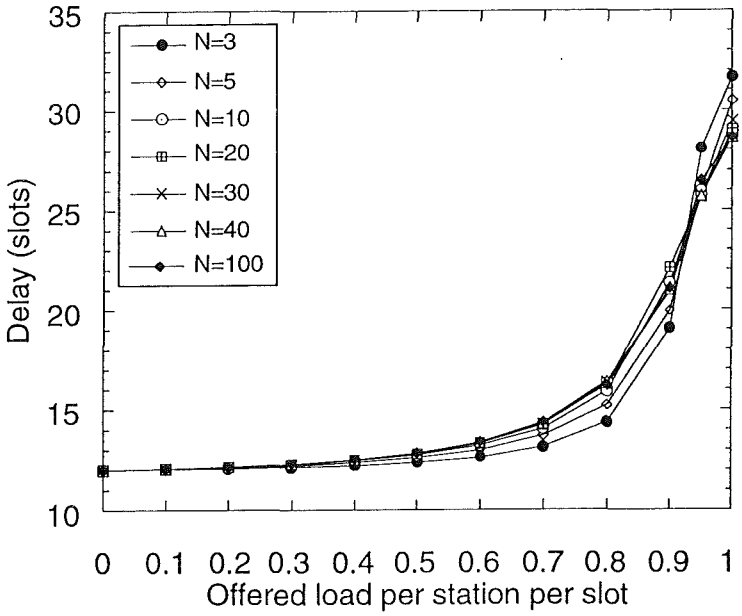


Figure 8.12: Mean packet delay of rcCA-FPCF* versus load, for varying number of stations. B=25, a=5. Relative precision $\leq$5%.

$Q_{i,2}$ be the second last stage, and so on. Denote the switch of $Q_{i,j}$ by $X_{i,j}$. At the output of the pipe lies an ON/OFF switch $X_{i,0}$.

Each stage has 2 inputs and one output. The output of $Q_{i,j}$ is connected to one of the inputs of $Q_{i,j-1}$ (the next stage down the pipe). The exception is stage $Q_{i,1}$. The output of this end stage is combined with signals from the output of $X_{i,0}$, and the merged signal enters the i-th port of the star coupler. The other input of $Q_{i,j}$, j=2, 3, ... , B, is connected to the output of $X_{i,j}$.

The delay loop of each stage has a length set such that the propagation delay through the loop of fiber equals one time-slot. The input to the Q_ith module is broadcasted to all $X_{i,j}$, j=1,2, ..., B+1. If $X_{i,j}$=ON, then the signal at its input is allowed to propagate to the input of $Q_{i,j}$. Otherwise it is absorbed.

Mapping the rcCA MAC Protocols onto an rcCA Implemented Using Optical Buffering

Under rcCA-FPCF/B and rcCA-FPCF/R defined in section 8.3.2, a packet placed in the j-th location of Q_i would be transmitted during the D(j,E)-th future time-slot, where

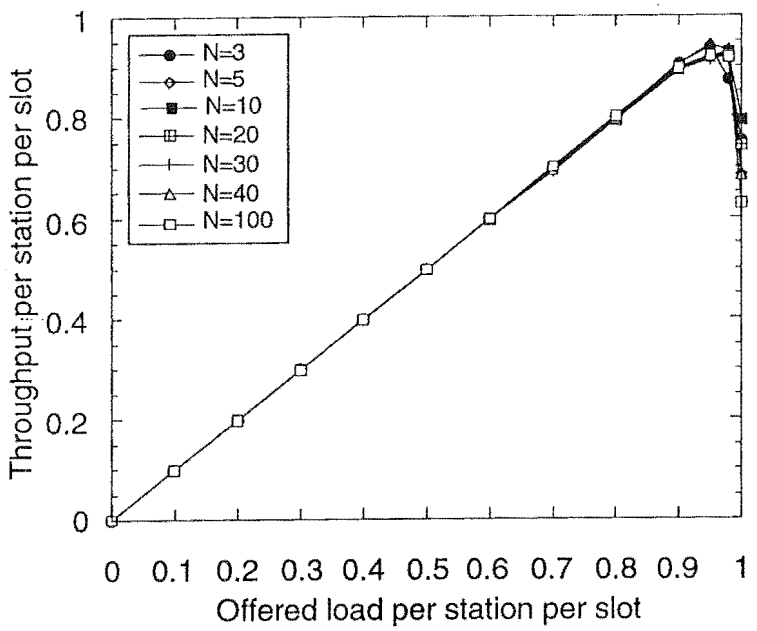


Figure 8.13: Throughput of rcCA-FPCF/R* versus load, for varying number of stations. B=25, a=5. Relative precision $\leq$5%.

$$D(j,E) = \begin{array}{ll} B+(E-j) & if\ (E-j)<0 \\ E-j & o.w. \end{array} \tag{8.1}$$

When optical buffers are used in place of electronic memory, an optical buffer module, Q_i, is accessed as follows. During the beginning of every time-slot t, all $X_{i,j}$, j=1,2, ..., B+1 are set to OFF by default. Suppose that FPCF$^*_{rc}$ decided that, during t, a packet from S_i needs to be rescued and placed in the j-th location of Q_i. Define L=D(j,E). Then the following step is actioned to re-schedule the arrival time of a packet to the L-th future time-slot :

- set $X_{i,L}$ to ON

No reception is necessary. No transmission is necessary. Note that if a packet would be successful (i.e. its destination would be free to receive it at its original time of arrival), then L=0, so $X_{i,0}$=ON, allowing the packet from S_i to propagate to the star coupler without delay. If $L>0$, $X_{i,L}$=ON and the packet would propagate through L stages. In effect, this schedules the packet for continuing from rcCA to its destination during the L-th future time-slot.

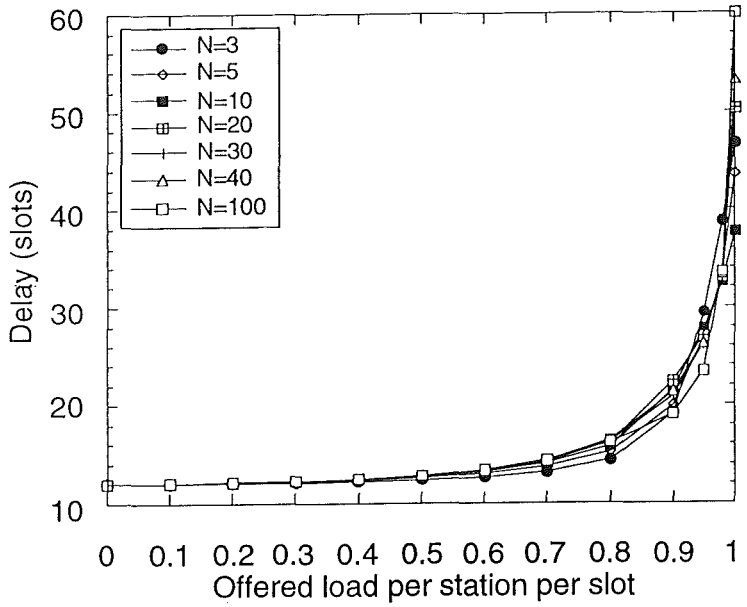


Figure 8.14: Mean packet delay of rcCA-FPCF/R* versus load, for varying number of stations. B=25, a=5. Relative precision $\leq$5%.

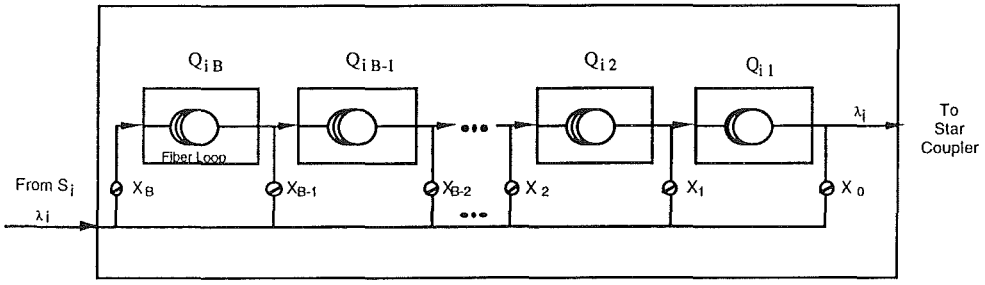


Figure 8.15: The configuration of an optical buffer module.

Technological Considerations

The proposed optical rcCA design has these costs :

1. B loops of fiber are needed for a buffer module with a capacity for storing B packets.

2. B+1 ON/OFF optical switches are needed for a buffer module of capacity B.

3. one photonic amplifier may be required per buffer module to compensate for the power loss from splitting the input signal into B+1 parts.

q	rcCA-FPCF/B		rcCA FPCF/R		
	Prob(Packet lost | q)		Prob(Packet reflected | q)		R_m
1.00	0.02397	(0.02381, 0.02413)	0.2034	(0.2033, 0.2034)	8.49
0.99	0.01724	(0.01711, 0.01737)	0.04216	(0.04175, 0.04257)	2.45
0.98	0.01157	(0.01148, 0.01166)	0.02844	(0.02821, 0.02866)	2.46
0.97	0.007151	(0.007096, 0.007206)	0.01901	(0.01887, 0.01916)	2.66
0.96	0.004003	(0.003964, 0.004043)	0.01154	(0.01146, 0.01162)	2.88
0.95	0.002065	(0.002046, 0.002084)	0.006304	(0.006249, 0.006360)	3.05
0.94			0.003009	(0.002982, 0.003035)	

Table 8.1: Multiplier effect of Reflection, measured by the ratio of the probability of packet loss (rcCA-FPCF/B) to the probability that a packet will be received by rcCA for Reflection (rcCA-FPCF/R), as a function of q. A N=10, B=40, a=5 network was assumed.

The proposed optical rcCA design has four advantages :

1. B+1 ON/OFF optical switches are the main components required for a buffer module of capacity B.

   An ON/OFF optical switch has much reduced functionality than a tuneable optical filter (used by tuneable receivers). Amongst the devices which can serve as an ON/OFF (transparent/opaque) switch are the magneto-optic effect light modulator currently in the market place, or the digital optical switch (see section 3.2.4 for a review). In fact, a dense 2-dimensional array of light modulators is a major component of some tuneable filters. The array of ON/OFF optical switches has been used as a rapidly reconfigurable diffraction grating to provide wavelength selectivity in a tuneable filter [WARR95]. It is also the key component of some optical interconnection networks [DIAS88].

   Since even a small value of B already yields near optimal rcCA-STAR performance, only a small number (B+1) of such ON/OFF optical switches are needed per buffer module. This suggests that the cost of the ON/OFF switches of a buffer module would be small, compared to the cost of a tuneable receiver.

Network	Computational Complexity	
	C_T	C_N
Request-schedule-then-transmit (CF-WDMA)	$20N^2 + 3N$	$20N^3 + 3N^2$
Detect-and-retransmit (DT-WDMA)	$22N + 15$	$22N^2 + 15N$
Request-schedule-then-transmit (DAS)	$\sum_{i=1}^{N}\sum_{j=1}^{N}(N-i+2)(N-j+2)(N+i-1)$	$N\sum_{i=1}^{N}\sum_{j=1}^{N}(N-i+2)(N-j+2)(N+i-1)$
Request-schedule-then-transmit (HTDM)	$\frac{M}{N+M}\sum_{j=1}^{N}\sum_{k=1}^{N}(N-j+2)(N-k+2)(N+j-1)$	$\frac{NM}{N+M}\sum_{j=1}^{N}\sum_{k=1}^{N}(N-j+2)(N-k+2)(N+j-1)$
sCA-Star	$8N^2 + 18N + 2$	$12N^2 + 21N + 2$
optCA-Star (optCA-MRS*)	$23N^2 + 25N$	$23N^2 + 29N$
optCA-Star (optCA-FPCF/B)	$3N + 2B(N-1) + 3$	$7N + 2B(N-1) + 3$
optCA-Star (optCA-FPCF/R)	$6N + 2BN + 2$	$11N + 2BN + 2$
rcCA-Star (rcCA-FPCF/B)	$5N + 2B(N-1) + 3$	$9N + 2B(N-1) + 3$
rcCA-Star (rcCA-FPCF/R)	$8N + 2BN + 2$	$13N + 2BN + 2$

Table 8.2: Comparison of the worst case computational complexity.

An alternative would be to use a $(B+1)$-way optical switch in place of $B+1$ ON/OFF optical switches. This option would avert the power loss from splitting the input signal into $B+1$ parts, but reduces the modularity of each stage.

2. Once a station transmits a packet, it will remain in the optical domain until it is received by its destination. Thus no data receivers or transmitters are needed at rcCA.

3. No (electronic) memory is needed at rcCA for storing "otherwise lost" packets.

4. Data signals propagating through rcCA (en route to ordinary stations) are still space division multiplexed.

   In WDM stars, packets may simultaneously arrive to ordinary stations on multiple WDM channels. Since signals propagating through rcCA are SDM, en route resolution of destination conflicts at rcCA using optical buffering is simpler than tasking all ordinary stations with conflict

resolution. Also, off the shelf photonic amplifiers can be used since all signals are on one wavelength. Working with SDM signals therefore bypasses the problems of gain equalisation which has to be considered when applying photonic amplification to WDM channels.

Notwithstanding, it should be pointed out that the electronic implementation of rcCA and optCA assumed heretofore is already optimal in terms of performance. If a packet needs rescuing by rcCA, it would not have been received by its destination during its original time-slot of arrival. Hence, rescued packets need to wait at least one time-slot before their destinations would be free. The total delay caused by E/O conversion at rcCA (when the buffer modules used electronic memory) is one time-slot, so the delay is desirable. The use of electronic memory by rcCA to store rescued packets therefore does not degrade the performance of rcCA-STAR networks. Nor does the use of electronic buffering limit the scalability of rcCA-STAR. Advances in technologies that allow the interface of ordinary stations to increase their transmission rate would enable rcCA to match the improved rate, since the technologies and the complexity of the busing structure of the interface of rcCA and ordinary stations are identical.

Optical buffering of packets is thus considered in this section only to provide an alternative implementation of rcCA-STAR, not to improve rcCA-STAR performance. The all optical rcCA-STAR alternative allows the engineer to choose an implementation based on costs and reliability considerations.

8.6 Conclusions

Whilst fiber has abundant bandwidth, for some WDM networks the utilization of channels is also of interest. For very large networks, photonic amplification of signals may be necessary. If photonic amplifiers are used within the star coupler, or if they are used to amplify signals beyond the star coupler, then the number of channels that can be simultaneously amplified is limited by the bandwidth of the photonic amplifiers and gain equalisers. The number of channels that can be used may also be limited by the tuning range of the tuneable receivers of stations. Previous architectures using the en route conflict resolution method required $2N$ data channels for a N station network.

This chapter investigated the rcCA-STAR architecture that can be implemented using only half the number of data channels than those studied previously. As a result, when the devices chosen for building the network

limits the number of channels (and hence the bandwidth) that can be used, the rcCA-STAR architecture can support twice the number of stations.

Results suggests that rcCA-STAR has these advantages

1. Near optimal mean packet delay and throughput (channel utilization) performances were obtained, provided that the offered load was below 98%.
2. Their MAC protocols have low time computational complexity when compared with the networks based on the request-schedule-then-transmit principle.
3. They require extremely simple logical buffer organisation.
4. Optical buffering can be easily implemented due to the use of rcCA buffer modules as a logical pipe with a constant flow rate.

rcCA-STAR has these drawbacks

1. Their throughput performance is slightly inferior compared to the optCA-STAR networks using optCA-FPCF/B (/R) protocols.
2. If optical buffer modules are used, then B+1 ON/OFF optical switches and B loops of fiber are required per buffer module.

Since the performance of rcCA-STAR is still near optimal, and since optical ON/OFF switches have much lower functionality than a tuneable filter (an array of such switches is a part of some tuneable receivers), these drawbacks are minimal. Thus one can conclude that rcCA-STAR is a favourable candidate for the implementation of WDM networks based on the en route conflict resolution principle.

Chapter 9

Comparison with Previous Solutions

In this chapter, networks in which the destination conflict resolution function is located at *all stations* and performed either *before* packet transmission (using the request-schedule-then-transmit concept) or *after* a destination conflict has been detected (using the detect-and-retransmit-if-lost concept), are compared with CA-STAR networks, in which only *one station* located at the entrance to the star coupler is responsible for detecting conflicts and optically buffering "otherwise lost" packets whilst they are *en route* to their destinations until their destinations are free to receive them (following the central placement concept).

The detect-and-retransmit class is represented by the DT-WDM (Dynamic Time WDM) networks [CHEN90], [PAPA92], [GREE93], while the request-schedule-then-transmit class is represented by the CF-WDM (Conflict-Free WDM) networks of [CHEN91], [CHEN92] and DAS-WDM (Dynamic Allocation Scheme WDM) and HTDM networks (Hybrid TDM) of [CHIP93]. DT-WDMA networks [CHEN90], [PAPA92], [GREE93] are based on the detect-and-retransmit principle. Accordingly, a station may send packets at will, then monitors the control channel to determine the outcome. If the station detects that its packet was lost due to a destination conflict, it retransmits the lost packet (see Fig. 1.4a for details). In CF-WDMA networks [CHEN91], [CHEN92], every station is required to establish a global view of all packets waiting for transmission in all stations, during every slot. During each time-slot, each station uses its record of the packets waiting at all stations, and the Maximum Remaining Sum scheduling algorithm to find a conflict free transmission schedule which specifies the packet transmis-

sions (by all stations) that should take place during the following slot (see Fig. 1.4b). Thus, CF-WDMA strictly prevents destination conflicts using the "request-schedule-then-transmit" method. DAS and HTDM follows a similar approach as [CHEN91], [CHEN92]. They employ a different conflict-free scheduling algorithm, and a novel aspect of HTDM is that it interleaves the "request-schedule-then-transmit" mode of operation with operations following a "static-conflict-free-transmission-schedule" to reduce the time computational complexity of the protocol, as reviewed in chapter 2.

CA-STAR networks are based on a "central placement" of the conflict resolution function, where function is performed by *only one station* (the CA) located at the entrance to the star coupler, and is performed whilst packets are already *en route* to their destinations. Networks based on this placement of the conflict resolution function are the sCA/B, sCA/R, optCA-MRS, optCA-FPCF/B, optCA-FPCF/R, rcCA-FPCF/B, and rcCA-FPCF/R networks. In deference to its later trait of resolving conflicts whilst packets are en route to their destinations, the CA-STAR networks can hereafter also be referred to as "en route" conflict resolution based networks.

Section 9.1 compares the electro-optical conversion overhead of the various networks. Section 9.2 compares the computation complexity of their MAC protocols. As mentioned the logical organisation of packets stored in the buffers within the network consumes resources (for recording the logical relationships) and processing power and time (for maintaining them as packets are added or transmitted). The complexity of the logical buffer organisation demanded by the various networks are compared in section 9.3. Section 9.5 compares their hardware demands. Finally, their throughput-delay characteristics are compared in section 9.6.

9.1 Electro-optical Conversions

When $a < 1$, the delay due to the electro-optical (E/O) conversions dominates mean packet delay. Table 9.1 compares the number of packet and mini-slot transmission(s) or reception(s) actioned for the successful exchange of one packet. Let Λ be the Random variable (R.V.) denoting the number of times that a packet is lost (DT-WDMA) and let $\Re$ be the R.V. denoting the number of Reflections (sCA/R, optCA-FPCF/R, rcCA-FPCF/R) experience by a packet, before it is received.

One can see that the E/O conversion overhead of the networks are similar, and their minimum overhead is identical. Furthermore, it should be noted that

Network	Packets		Mini-slots	
	Min.	Max.	Min.	Max.
Detect-and-retransmit (DT-WDMA)	2	2+Λ	2N+1	2N+1+Λ(2N+1)
Request-schedule-then-transmit (CF-WDMA, DAS, HTDM)	2	2	N+1	N+1
CA-STARs in which CA's buffers are Impemented Using Electronic Memory :				
sCA-Star (sCA-A)	2	2+2	2N+2	2N+2
optCA-Star (optCA-MRS*)	2	2+2	N+3	N+3
optCA-Star (optCA-FPCF/B)	2	2+2	N+3	N+3
optCA-Star (optCA-FPCF/R)	2	2+4$\Re$	N+3	N+3+(N+3)$\Re$
optCA-Star (rcCA-FPCF/B)	2	2+2	N+3	N+3
optCA-Star (rcCA-FPCF/R)	2	2+4$\Re$	N+3	N+3+(N+3)$\Re$
CA-STARs in which CA Uses Optical Delay Lines to Reschedule Packet Arrival times :				
optCA-Star (optCA-FPCF/B)	2	2	N+3	N+3
optCA-Star (optCA-FPCF/R)	2	2+2$\Re$	N+3	N+3+(N+3)$\Re$
optCA-Star (rcCA-FPCF/B)	2	2	N+3	N+3
optCA-Star (rcCA-FPCF/R)	2	2+2$\Re$	N+3	N+3+(N+3)$\Re$

Table 9.1: Comparison of the number of electro-optic conversions needed for the successful delivery of a packet.

the figures given for CA-STAR networks represent a kind of worst case analysis. In CA-STAR networks implemented using the optCA or rcCA, only the CA station needs to receive and process N mini-slots per time slot. Ordinary stations have to receive just one mini-slot.

In CF-WDMA, a packet exchange requires *exactly* 2 E/O packet conversions, plus N+1 mini-slot E/Os. In DT-WDMA, the minimum overhead are 2 E/O packet conversions, plus N+1 mini-slot E/Os by the destination station, and N+1 mini-slot E/Os by the source for success deduction. A packet is delayed by Λ extra E/O conversions for packet re-transmission, and $\Lambda(2N+1)$ extra E/O conversions of mini-slots (Each loss of a packet requires one mini-slot transmission for (re)transmission signalling, N mini-slot receptions by the destination station, and N mini-slot receptions by the source station for deducing the outcome of the retransmitted packet). Λ=0,1,2, ...

With sCA/R, optCA-FPCF/R, and rcCA-FPCF/R, the minimum overhead are 2 E/O packet conversions, plus N+3 mini-slot E/O conversions (signalling, reception of N mini-slots and signalling by optCA, and one mini-slot

reception by destination). If optical delay lines are used for re-scheduling the arrival times of packets involved in a conflict, then packets remain in the optical domain once transmitted by their source stations, even if they were involved in a destination conflict. If electronic memory is used instead, then if a packet is involved in a destination conflict, it is delayed by *at most* 2 extra packet E/O conversions (reception into optCA, and transmission from CA) . The overhead of sCA/R, optCA-FPCF/R, and rcCA-FPCF/R is greater by $4\Re$ packet and $(N+3)\Re$ mini-slot E/O conversions, if CA used electronic memory, $\Re$=0,1,2, ... Typically E[$\Re$]<<1, For example, see Table 7.1.

Packet and mini-slot transmissions and receptions may be pipelined, or performed in parallel. For instance, some mini-slot O/E conversions occur during one slot, whilst others occur during several time-slots. Also, when a >0, the delay of one time-slot for packet transmission can overlap with propagation and/or reception. For concreteness, suppose a=0.5. Then the source-to-destination propagation delay equals one time-slot. The minimum packet delay due to E/O conversions is one time-slot for packet transmission, plus one time-slot for packet reception. The *total* minimum packet delay equals propagation delay (one time-slot) plus E/O conversion delay (two time-slots) minus one time-slot. One time-slot should be subtracted to account for the fact that the E/O delay for packet transmission overlaps with propagation delay during the slot when transmission occurs.

Thus the delay of a packet due to E/O conversions may not equal the number of E/O conversions actioned for its successful transfer, even when a is small. Table 9.2 shows the part of the total delay of a packet that is caused by the E/O conversions and propagation when a=0.5. One can see that when a=0.5, the packet delays in the networks caused by E/O conversions are similar, and that the minimum overhead of E/O conversions and propagation is identical. It is important to note that the delay of one time-slot introduced by E/O conversion at CA (assuming that electronic buffer modules are used instead of optical buffers) is desirable (indicated in the table by underlining), because if a packet needed to be rescued by CA (whether its is of the sCA, optCA, or rcCA design), then its time of arrival to its destination needed to be delayed by at least one time-slot. Therefore there is no delay overhead from en route conflict resolution using CA, unless Reflection is used. The delay overhead from Reflection (case of CA-STAR networks using the /R type of protocols) is $4\Re$ time-slots. Typically E[$\Re$]<<1, see Table 7.1.

Network	Delay (time-slots) Min.	Max.
Detect-and-retransmit (DT-WDMA)	3	3+2Λ
Request-schedule-then-transmit (CF-WDMA)	5	5
CA-STARs in which CA use electronic buffers to store "otherwise lost" packets :		
sCA-STAR (sCA/B)	3	3+$\underline{1}$
optCA-Star (optCA-MRS*)	3	3+$\underline{1}$
optCA-Star (optCA-FPCF/B)	3	3+$\underline{1}$
optCA-Star (optCA-FPCF/R)	3	3+$\underline{1}$+4ℜ
optCA-Star (rcCA-FPCF/B)	3	3+$\underline{1}$
optCA-Star (rcCA-FPCF/R)	3	3+$\underline{1}$+4ℜ
CA-STARs in which CA Uses Optical Delay Lines to Reschedule Packet Arrival times :		
optCA-Star (optCA-FPCF/B)	3	3
optCA-Star (optCA-FPCF/R)	3	3+2ℜ
optCA-Star (rcCA-FPCF/B)	3	3
optCA-Star (rcCA-FPCF/R)	3	3+2ℜ

Table 9.2: Comparison of delay per packet measured in time-slots, caused by optical/electronic conversions and propagation; a=0.5.

9.2 Comparison of Computational Complexity

Define *network computational complexity* (C_N) as the maximum number of scalar operations for MAC purposes that is performed in the network during one time slot. Following [CHEN91], [CHEN94], an assignment, comparison, addition, or subtraction will be considered as a scalar operation.

Define *time computational complexity* (C_T) as the maximum number of scalar operations for MAC purposes performed during one time slot by the station which has the most complex MAC procedure. If we are concerned about the MAC protocol's electronic processing operations creating a bottleneck, then C_T may be a more suitable complexity index.

Expressions for the C_N and C_T electronic processing complexities of the sCA/B, sCA/R, and optCA-MRS networks, and that of networks using the

request-schedule-then-transmit or the detect-and-retransmit-if-lost schemes, are derived in Appendix F. The C_N and C_T of optCA-FPCF/B optCA-FPCF/R and rcCA-FPCF/B rcCA-FPCF/R were derived in sections 7.3.4 and 8.4.4 respectively. The results are summarised in Table 9.5. The comparison is based on a network with N stations, each with a transmit buffer of size B. As shown in column three, optCA-FPCF/B(/R) and rcCA-FPCF/B(/R) enjoys considerably lower C_N (O(N)) than the other networks ($O(N^5)$ for DAS and HTDM (assuming that the fraction of open slots in a frame remains constant) [CHIP93], $O(N^3)$ for CF-WDMA [CHEN91], and $O(N^2)$ for DT-WDMA [CHEN90]). The CA-STAR networks obtain low C_N by complexity inversion: MAC tasks that burden all stations in the other architectures (e.g. deciding which packet to receive) are performed by just one station, i.e. by optCA. In this way, much replication of electronic processing in previous architectures has been eliminated. Ordinary stations need to process only one mini-slot during every time slot. In contrast, previous solutions [CHEN90], [CHEN91], [CHEN92], [CHLA91], [CHLA94], require all stations to process information in all mini-slots during every time-slot.

Expressions for C_T are given in column two. The CA-STAR networks differs from the other networks in that the MAC of their ordinary stations has lower computational complexity than that of the CA station. Thus we represent the time complexity of the CA-STAR networks by the complexity of their protocols for CA. One can find that the optCA-FPCF/B, optCA-FPCF/R, rcCA-FPCF/B, and rcCA-FPCF/R protocols have lower order C_T than the protocols executed by stations in the DAS ($O(N^4)$) and HTDM ($O(N^4)$) networks [CHIP93], and the CF-WDMA ($O(N^2)$) networks [CHEN91], as well as that of the CA MAC protocol of optCA-MRS ($O(N^2)$). The optCA-FPCF/B (/R) and rcCA-FPCF/B (/R) protocols have linear, i.e. O(N) complexity; the same as DT-WDMA [CHEN90]. Recall that CA-STAR's performance remained near optimal for a fixed B, as network size increased. Thus we can treat B in the expressions for the computational complexity of optCA-FPCF/B (/R) and rcCA-FPCF/B (/R) networks as a constant, even as N is increased.

Almost all operations of the FPCF based CA-STAR protocols are matrix operations with potential parallelism that can be exploited by execution on a simple vector or multiprocessor system. The execution time of the FPCF based protocols can thus be further reduced. Also, many of the comparisons can be performed in parallel using associative testing. Since in optCA-STAR (rcCA-STAR) networks only one station - the optCA (rcCA) - is tasked with destination conflict resolution, vector or multiprocessing hardware supplied to just one station can speedup the protocol (electronic) execution time of the en-

Network	Computational Complexity	
	C_T	C_N
Request-schedule-then-transmit (CF-WDMA)	$20N^2 + 3N$	$20N^3 + 3N^2$
Detect-and-retransmit (DT-WDMA)	$22N + 15$	$22N^2 + 15N$
Request-schedule-then-transmit (DAS)	$\sum_{i=1}^{N}\sum_{j=1}^{N}(N-i+2)(N-j+2)(N+i-1)$	$N\sum_{i=1}^{N}\sum_{j=1}^{N}(N-i+2)(N-j+2)(N+i-1)$
Request-schedule-then-transmit (HTDM)	$\frac{M}{N+M}\sum_{j=1}^{N}\sum_{k=1}^{N}(N-j+2)(N-k+2)(N+j-1)$	$\frac{NM}{N+M}\sum_{j=1}^{N}\sum_{k=1}^{N}(N-j+2)(N-k+2)(N+j-1)$
sCA-Star	$8N^2 + 18N + 2$	$12N^2 + 21N + 2$
optCA-Star (optCA-MRS*)	$23N^2 + 25N$	$23N^2 + 29N$
optCA-Star (optCA-FPCF/B)	$3N + 2B(N-1) + 3$	$7N + 2B(N-1) + 3$
optCA-Star (optCA-FPCF/R)	$6N + 2BN + 2$	$11N + 2BN + 2$
rcCA-Star (rcCA-FPCF/B)	$5N + 2B(N-1) + 3$	$9N + 2B(N-1) + 3$
rcCA-Star (rcCA-FPCF/R)	$8N + 2BN + 2$	$13N + 2BN + 2$

Table 9.3: Comparison of the worst case computational complexity.

tire network. Economically, optCA-STAR and rcCA-STAR is best positioned to benefit from the use of SIMD or MIMD hardware.

9.3 Comparison of Buffer Organisation Complexity

Physical buffer organisation is defined by the electronic components of a station's buffer, and the access paths they support. A typical buffer contains a memory, a time multiplexed read address/data bus, a write address/data bus, and several control and power lines. The memory is an array of addressable storage elements called words. For convenience, divide the memory into B packet size addressable locations, each of which is made of a fixed number of words. The packet size locations have unique addresses, and are identical in storage characteristics.

On top of this physical buffer organisation, we can impose a particular *logical buffer organisation (logical structure)*, that defines relationships between the packets stored in the physical buffer. Common logical buffer organisation of packets are the FIFO (First-In-First-Out) queue, LDF (Largest accumulated packet Delay served First) queue [CHLA91], and multi-queues such as N FIFO queues of packets to be formed in each physical buffer).

A logical buffer structure is typically *created* using pointers which record the relationship between packets stored in a given physical buffer. A logical buffer organisation is usually *maintained* by updating pointers when packets are added/transmitted to the physical buffer. Imposing a logical buffer organisation on top of a physical buffer organisation thus has resource (pointers) and processing time (updating pointers) overhead.

The logical structures that need to be maintained for each buffer for the execution of DAS, HTDM, CF-WDMA, DT-WDMA, sCA-STAR, optCA-STAR (optCA-MRS*, optCA-FPCF and optCA-FPCF/R), and rcCA-STAR (rcCA-FPCF/B and rcCA-FPCF/R) networks are compared in Table 9.4.

9.3.1 DT-WDMA

Packets in the transmit buffer of each DT-WDMA station are logically organised into a special queue. Each new packet is tagged with its arrival time and a status flag which is initialised to *waiting*. The packet in the transmit buffer queue which 1) has status *waiting* and 2) has the earliest arrival time *amongst the waiting packets*, is chosen for transmission. Hence packets are chosen according to a "conditional-FIFO" discipline. The transmitted packet has its status changed to *outstanding*. Whenever a packet loss is detected, the corresponding queued packet is located and its status changed from *outstanding* back to *waiting*.

9.3.2 CF-WDMA, DAS and HTDM

The CF-WDMA protocol needs N -1 FIFO queues of packets to be formed in the transmit buffer of each station. In the case of DAS and HTDM networks [CHIP93], it was assumed that packets at each station are organised into N-1 FIFO queues by storing packets in one of N-1 separate FIFO buffers. The purpose of these logical structures can be seen by considering the method used by their stations for deciding which packets to transmit during a time slot. In CF-WDMA, DAS and HTDM networks, during each time-slot all stations

execute the same algorithm to produce the same transmission schedule. A transmission schedule specifies which station is allowed to transmit a packet to a given destination. If the schedule allowed station S_i to transmit a packet to S_j, then S_i must :

1. find all packets in its transmit buffer that are destined for S_j, and
2. of the packets destined for S_j, choose the one that was the first to be generated for transmission.

To facilitate this search, packets in the transmit buffer of each station can be logically organised into N -1 FIFO queues, where packets in logical queue j are destined for station S_j. If the schedule allowed station S_i to transmit to S_j, S_i can identify the packet it should transmit as the one in the head of the j-th logical queue. This is the logical buffer organisation used by CF-WDMA.

Alternatively, each station can be provided with N-1 physical buffers. A new packet at station S_i which is destined for S_j would be stored in the j-th physical buffer of S_i. Packets in each physical buffer are logically organised into a FIFO queue. If the schedule allowed station S_i to transmit to S_j, S_i can identify the packet it should transmit as the one in the head of the logical FIFO queue in the j-th physical buffer. This buffer organisation was assumed in DAS and HTDM networks.

9.3.3 sCA-STAR using the sCA/A or sCA/R Protocols

No logical buffer organisation is needed at the transmit buffer of ordinary stations of a sCA-STAR network, since if a packet is generated during t, its destination address is signalled during t, and it would be transmitted unconditionally during t+1. Thus, there are at most 2 packets in the transmit buffer. During a time-slot, the station can deterministically (D) access one of the two buffer spaces for storing a newly generated packet (if any) and access the other for transmitting the signalled packet (if any). The roles of the two physical spaces can be rotated after each time-slot.

However, packets in the buffer of sCA have to be logically organised into N FIFO queues. In addition, a fixed number of physical buffer locations is dedicated for use by tempbuff.

9.3.4 optCA-STAR using the optCA-MRS* Protocol

No logical buffer organisation is needed at the transmit buffer of ordinary stations of optCA-MRS* networks too. There can be at most 3 packets in the transmit buffer, and the three buffer locations can be accessed deterministically.

Packets in each buffer module of optCA are logically organised into N-1 FIFO queues. According to optCA-MRS*, the optCA station executes the MRS* scheduling algorithm during t to determine which buffer module is allowed to transmit a previously-rescued packet to a given destination. If the schedule allowed buffer Q_i to transmit to S_j, optCA can identify the packet it should transmit as the one in the head of the j-th logical queue in buffer Q_i.

9.3.5 optCA-FPCF/B, optCA-FPCF/R, rcCA-FPCF/B, and rcCA-FPCF/R Networks

As shown in Table 9.4, the opt and rcCA-FPCF/B (/R) networks have the simplest logical buffer organisation of the networks considered: *no logical structures in optCA's (or ordinary stations') buffers need to be maintained.*

This improvement comes from the fact that FPCF uses the knowledge of future transmissions to streamline buffer organisation. Under opt or rcCA-FPCF/B (/R), all packets in optCA (rcCA) that are scheduled for transmission during a time slot are simply identified by the E-th column of C. Namely, if physical location $C_{iE} \neq$ empty, then the packet in the C_{iE} should be transmitted, i=1, ..., N. The column index E rotates *deterministically* after each slot. Since the buffer locations of all packets that are scheduled for transmission are completely identified by the column index E (whose value deterministically increments by one, modulo B, after each time-slot), there is no need to search inlet buffers for the locations of the packets that are scheduled for transmission. Thus, no queues nor lists of packets need to be created nor maintained. No logical relationships between packets in a buffer needs to be recorded. Planning the future transmission of a packet becomes equivalent to determining the physical address for storing a packet. Logical buffer organisation is therefore completely subsumed by the Forwarding Planning of packet transmissions. Furthermore, the FIFO service order is naturally maintained.

No logical buffer organisation is needed at the transmit buffer of ordinary stations of opt or rcCA-FPCF/B (/R) networks too. There can be at most 3 packets in the transmit buffer, and the three buffer locations can be accessed

deterministically.

9.4 Buffer Access Modes

Finally, with opt and rcCA-FPCF/B (/R), buffers are accessed using relatively simple access modes (Table 9.4). For input to optCA, random access (**R**) write is needed, but reading is done using column access mode (**C**) which permits somewhat simpler addressing. In ordinary stations, access is deterministic with the locations for reading and writing rotating during every slot.

Network	Buffer access modes		Logical Structures Maintained
	Read	Write	
Request-schedule-then-transmit (DAS, HTDM)	R	R	N-1 FIFO buffers per station
Request-schedule-then-transmit (CF-WDMA)	R	R	N-1 FIFO Queues per station
Detect-and-retransmit (DT-WDMA)	R	R	1 FIFO transmission and 1 Outstanding (retransmission) queue
sCA-STAR Networks :			
sCA/B and sCA/R (Ordinary Stations)	D	D	None required
sCA/B and sCA/R (sCA Station)	R	R	N FIFO queues and 1 list (tempbuff)
optCA-STAR Networks Using MRS* :			
optCA-MRS* (Ordinary Stations)	D	D	None required
optCA-MRS* (sCA Station)	R	R	N FIFO queues
optCA-STAR and rcCA-STAR Networks Using FPCF*:			
opt and rcCA-FPCF/B (/R) (Ordinary Stations)	D	D	None required
opt and rcCA-FPCF/B (/R) (opt or rcCA Station)	D	R	None required

Table 9.4: Buffer organisation complexity, and buffer access modes of various WDM networks

9.5 Comparison of Hardware Demand

Table 9.5 compares the hardware demand of WDM networks based on various destination conflict resolution methods. The comparison is based on the requirements per station in an N station network. *The values for CA-STAR*

accounts for the hardware requirements of the CA, divided amongst N stations. One can see that the hardware and bandwidth demand per station of optCA-STAR and rcCA-STAR is comparable with the detect-and-retransmit (DT-WDMA), detect-and-retransmit with multiple reception opportunities (SDL), and request-schedule-then-transmit networks (CF-WDMA, DAS and HTDM), and less than those based on "receiver replication".

Nevertheless, it should be noted that the "receiver replication" networks require only fixed tuned transmitters/receivers, so they may be more economical if the cost of the fixed tuned components is considerably lower than rapidly tuneable ones, when N is not too large.

Network	Transmitter	Receiver	Max. buffer module bandwidth (pkts/slot)	Min. Buffer stay per packet
Detect *after* conflict occured, and retransmit (DT-WDMA).	2 FT	1 FF, 1 TF, 2R	2	2a+1 slots
Request and schedule *prior* to packet transmission (CF-WDMA).	2 FT	1 FF, 1 TF, 2R	2	2a+1 slots
Multiple tunable switches and optical delay lines per station (SDL).	2 FT	1 FF, d+1 TF, 2R	2	2a+1 slots
Multiple Receivers per station (LambdaNet).	1 FT	NR demux	2 (assuming N receiver buffers per station)	1 slot
En route conflict resolution (sCA-STAR under sCA/B and sCA/R).	3 FT	1 FF, 1 TF, 3R	N	1 slot
En route conflict resolution (optCA-STAR in which optCA use electronic memory to store rescued packets)	3 FT	1 FF, 1 TF, 3R	2	2 slots
En route conflict resolution (optCA-STAR in which optCA use optical delay lines to reschedule "otherwise lost" packets)	2 FT	1 FF, 1 TF, 2R, (B+1)S	2	2 slots
En route conflict resolution (rcCA-STAR in which rcCA use electronic memory to store rescued packets)	3FT	1 FF, 1 TF, 3R, 1S	2	2 slots
En route conflict resolution (rcCA-STAR in which rcCA use optical delay lines to reschedule "otherwise lost" packets)	2 FT	1 FF, 1 TF, 2R, (B+2)S	2	2 slots

Table 9.5: Comparing the Hardware Demands of WDM networks based on various destination conflict resolution methods. FT denotes fixed transmitter, FF denotes fixed tuned filter, TF denotes tunable filter, R denotes RF receiver, and S denotes ON/OFF optical switch. Figures for CA-STAR accounts for requirements per station of CA.

9.6 Performance Comparison

When a >1, propagation delay becomes a significant determinant of the network's delay performance. The throughput-delay characteristics of the various CA-STAR networks when a=5 are compared in Fig. 9.1.

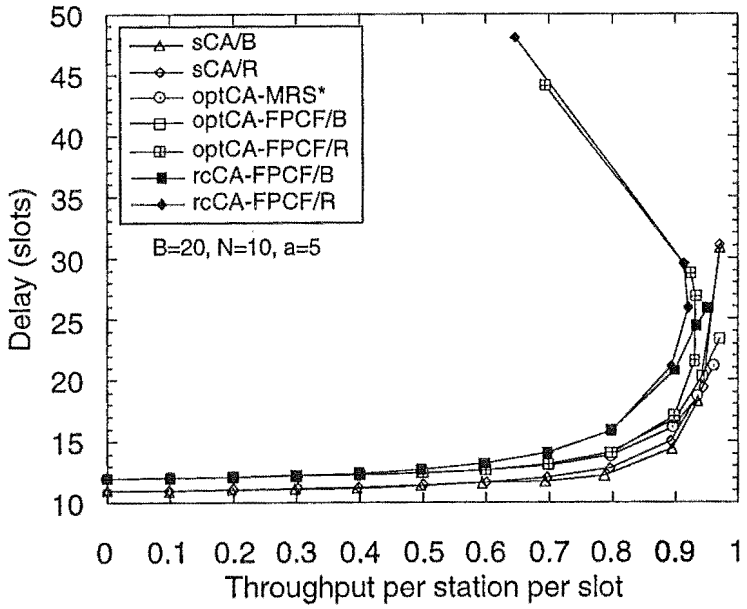


Figure 9.1: Comparison of throughput-mean delay characteristics of various CA-STAR networks. N=10, B=20, a=5. Relative precision: $\leq$5%.

One can see that all CA-STAR networks show near ideal throughput-delay characteristics, provided that the working throughput is below 95%. Above 95%, the protocols that employ Reflection (to provide guaranteed delivery of packets) exhibit significantly higher performance degradation. This is probably due to the fact that a packet is delayed by $2a$+2 extra time-slots each time it is reflected. Fortunately, even when B is small (e.g.B=10), reflection is a rare event, except at very high offered load, for example see Table 9.1. One can also observe that although the rcCA-STAR networks are somewhat inferior to the other CA-STAR networks, the difference in performance is significant only at very high offered load ($\geq$ 95%). As mentioned, this is due to the fact that rcCA-STAR networks are implemented using half of the number of channels (bandwidth) required by the other CA-STAR networks. To be on the conservative side, we shall use rcCA-STAR protocols to represent the CA-STAR networks in the following comparison.

The throughput-delay characteristics of the rcCA-STAR networks (en route conflict resolution) are compared to the DT-WDMA (detect-and-retransmit),

and CF-WDMA (request-schedule-then-transmit) networks in Fig. 9.2, assuming N=10, B =20, and a=5. The CA-STAR networks, represented by rcCA-FPCF/B and rcCA-FPCF/R, show a very good performance. rcCA-STAR networks can achieve a maximum throughput (i.e. maximum channel utilisation) of 92% (rcCA-FPCF/R) to 98% (rcCA-FPCF/B), compared to 62% for DT-WDMA. One can observe that the rcCA-STAR networks using the rcCA-FPCF/B protocol gives somewhat higher throughput than the network following the "request-schedule-then-transmit" method (CF-WDMA). This may be due to the fact that the FPCF algorithm has slightly higher efficiency than the MRS algorithm (see section 6.4). The other reason for the improved performance of the rcCA-STAR networks is that the traffic matrix maintained by stations in the CF-WDMA networks for schedule computation would be $2a$+1 slots outdated (time for request propagation). Thus the transmission schedules computed by stations in CF-WDMA may be based on out-of-date information when a >1.

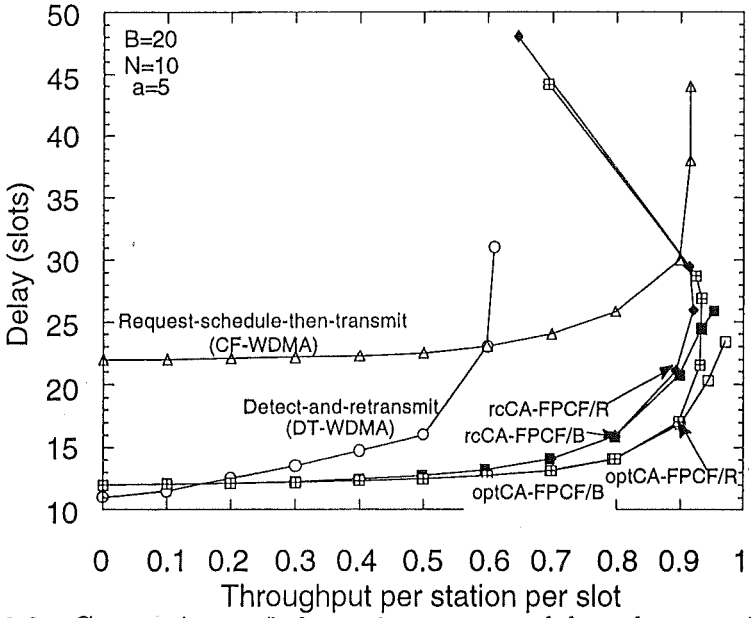


Figure 9.2: Comparison of throughput-mean delay characteristics of rcCA-STAR ("en route" conflict resolution using rcCA-FPCF/B and rcCA-FPCF/R protocols), with the "detect-and-retransmit" (DT-WDMA) and "request-schedule-then-transmit" (CF-WDMA) networks. N=10, B=20, a=5. Relative precision $\leq \pm 5\%$.

The most significant advantage of CA-STAR is its lower average packet delay, and it could be even more evident if a larger network diameter was assumed. The lower delay could be explained since packets in rcCA-FPCF/B experience a pre-transmission delay of just two slots (see Fig. 1.4c), versus

$2a+1$ slots for CF-WDMA (Fig. 1.4b). Packets in DT-WDMA encounter a delay penalty of at least $2a+1$ slots (one source-destination propagation delay period, plus one slot for signalling) if lost due to a destination conflict (Fig. 1.4a). In contrast, packets in rcCA-STAR typically experience no extra propagation penalty, even if they are involved in a destination conflict. They would be delayed only until their destination is free to receive them (Fig. 1.4c). The exception is if a packet is reflected, which is a rare event, except at very high offered load (see Table 8.1).

9.7 Comparison of Fault Tolerance

A feature of detect-and-retransmit-if-lost and request-schedule-then-transmit networks is that they are intrinsically tolerant of failures in their conflict-resolution hardware : all N stations are equipped with components for independently computing the same conflict-free transmission schedule or for deducing conflicts that occurred and determining which packets to receive and (re)transmit. Thus the failure of the conflict-resolution module in one station allows communication between other stations to proceed unaffected. In CA-STAR networks, only one set of components for conflict-free schedule computation (placed at CA) is required for the entire network, but communication services to all stations are affected if it fails. Refer to a set of conflict resolution hardware as a conflict resolution *module*.

This section is concerned with comparing quantitatively networks based on the above mentioned placements in terms of their tolerance of conflict resolution hardware failures. In addition, a variant of rcCA-STAR where $M \leq N$ conflict-resolution modules are located at rcCA, operating in a hot redundancy mode, will be considered. Unlike request-schedule-then-transmit networks, the replication of conflict-free-scheduling hardware is not intended to give more stations the ability to perform the conflict-resolution task, but are introduced to protect rcCA from faults in its conflict resolution hardware. rcCA fails to perform the conflict-resolution function only if all M conflict-resolution modules fail. These alternative placements of the conflict-resolution function will be referred to as :

1. Distributed replicated placement (detect-and-retransmit and request-schedule-then-transmit).
2. Central placement (represented by rcCA-STAR networks).

3. Central replicated placement, a variant of rcCA-STAR where M≤N conflict-resolution modules are located at rcCA, operating in a hot redundancy mode.

9.7.1 Fault model

In order to obtain specific, quantitative results on the effect of the placement of the destination conflict resolution function on the network's fault tolerance, we shall make the following assumptions regarding the characteristics of faults which occur.

M1) The only faults that occur are those caused by the failure of conflict resolution modules.

M2) Each conflict resolution module can be either fully operational or failed.

M3) The lifetime (L) of a conflict resolution module is a continuous random variable with a probability density function of the form

$$f_L(t) = \begin{cases} \frac{1}{\lambda} e^{-t/\lambda} & \text{for } t > 0 \\ 0 & \text{o.w.} \end{cases}$$

where l is assigned a numerical value equal to its expected lifetime[1]. The specific algorithm executed by a conflict resolution module depends on the type of network; we do not distinguish between them in our fault model (i.e. l is assumed to be the same for conflict-resolution modules of all networks), and therefore do not account for the lower computational complexity of the algorithm executed by the rcCA of rcCA-STAR networks. An algorithm with lower complexity and which requires simpler logical buffer organisation reduces hardware demand, with consequent improvements in module lifetime. Similarly, the rcCA-STAR networks and some detect-and-retransmit networks use optical delay lines in their conflict resolution module(s) instead of electronic memory. Although a central placement may be realised using electronic memory [YAU96a], [YAU96b] (electronic memory allows the use of more complex conflict resolution procedures since the duration of stay of a packet does not have to be bounded [YAU96b]), they are likely to have shorter expected lifetimes, and their access speed may become the network's bottleneck. We do not account for the improvement in expected lifetime from the use of optical memory (delay lines).

M4) The conflict-resolution modules in a network operate independently: the failure of one does not affect the performance characteristics of others.

---

[1]Empirical studies have shown that electronic equipment lifetimes often have a probability function of the exponential or geometric (discrete case) form.

Among the more significant non-conflict resolution module faults not modelled are problems such as fires, earthquakes, blackouts, and floods. It must be noted that using a central placement, even the provision of redundant conflict resolution hardware at rcCA would not protect it against these threats : a fire, earthquake, blackout, or flood at the star coupler (site of rcCA) is likely to affect all conflict resolution modules located there (invalidating M4), thereby causing a total failure. Indeed, one property of detect-and-retransmit or request-schedule-then-transmit networks is that a fire, earthquake, blackout, or flood at any one location would not affect the network access of the remainder, provided that the star coupler can be protected.

9.7.2 Fault Tolerance Measures

The aim is to compare the three placements in terms of 1) the expected number of stations which can communicate using the network after the network had been operating for a period of *t* time units, called the *availability* of communication service at *t*, A(t), and 2) the *uncertainty regarding the availability* of service at time t, called the *riskinesss* of a placement, R(t). If Y(t) denotes the number of stations with network access after t time units, then

$$A(t) = E[\text{number of stations with network access after t time units}]$$

$$= E[Y(t)], \quad \text{and}$$

$$R(t) = Var[Y(t)]$$

The larger the R(t), everything else remaining constant, the greater the dispersion of A(t) and the greater the uncertainty of network availability or risk of the placement. Given a choice between two placements that offer the same expected network availability (A(t)), the placement with the smallest R(t) is superior from the fault tolerance viewpoint since it has relatively smaller risk.

9.7.3 Comparison

Using the assumptions of the fault model, we can easily obtain solutions to $A_i(t)$ and $R_i(t)$, i=1,2,3 for networks based on the three types of placements of the destination conflict resolution function.

1) Detect-and-retransmit and Request-schedule-then-transmit Networks (Distributed Replicated Placement) :

The expected number of stations with network access after t time units can be obtained by noting that P(a given station can communicate at time t) is 1-P(the lifetime of that station is less than t). $A_1(t)$ is then

$$A_1(t) = \sum_{i=1}^{N} 1 - \int_0^t \frac{1}{\lambda} e^{-x/\lambda} dx$$

$$= N - N(1 - e^{-\lambda t}) \qquad (9.1)$$

Let $X_i(t)=1$ if station S_i has network access at time t, $X_i(t)=0$ o.w.. Assuming that the failure of one station's conflict resolution module does not affect the performance characteristics of others (M4), $R_1(t)$ can be written as

$$R_1(t) = \sum_{i=1}^{N} Var(X_i(t))$$

$$= N (E[X^2(t)] - E[X(t)]^2)$$

$$= N\left([1 - (1 - e^{-\lambda t})] - [1 - (1 - e^{-\lambda t})]^2\right) \qquad (9.2)$$

which is the same as the variance of $X_i(t)$ multiplied by the number of stations.

2) Central Placement (rcCA-STAR)

All stations are functional when the destination conflict resolution module of rcCA is operational. Otherwise the entire network fails. The expected number of stations with network access is therefore the probability that rcCA can perform the conflict resolution function times the number of stations,

$$A_2(t) = N\, P(\text{conflict resolution module of rcCA operational at time t})$$

$$= N - N(1 - e^{-t\lambda}). \qquad (9.3)$$

The riskiness of the central placement is obtained from the variance of the probability that the conflict resolution module of rcCA is operational at time t, after scaling by the constant N,

$$R_2(t) = N^2\left(1 - (1 - e^{-t\lambda})\right)\left(1 - (1 - (1 - e^{-t\lambda}))\right) \qquad (9.4)$$

Clearly, even though it demands only one conflict resolution module, a central placement of the conflict resolution function enjoys the same availability as a distributed replicated placement (detect-and-retransmit and request-schedule-then-transmit) since $A_1(t) = A_2(t)$. One can also conclude that the central placement has N times the risk of a distributed replicated placement, since $R_2(t) = N\, R_1(t)$. The availability in a network using a central placement can vary abruptly from N to zero, whereas the same measure for a network based on a distributed replicated placement decreases incrementally (graceful degradation).

3) Central Replicated Placement

In a rcCA-STAR network with M-1 redundant conflict resolution modules (M≤N), all of which are located at rcCA, Y(t)=N if at least one module is functional after t time units, Y(t)=0 o.w.; thus

$$A_3(t) = N\,P(\text{at least one module fault-free after t time units})$$

$$= N\left(1-(1-e^{-T\lambda})^M\right) \qquad (9.5)$$

The riskiness of this placement is the variance of Y(t). Noting that Var(Y(t)= $E[Y^2(t)] - E[Y(t)]^2$, and integrating and substituting gives

$$R_3(t) = N^2\left(1-(1-e^{-t\lambda})^M\right) - \left(N\left(1-(1-e^{-T\lambda})^M\right)\right)^2. \qquad (9.6)$$

Fig.9.3 shows the network availability obtained using the various placements of the destination conflict resolution function versus M (case of central replicated placement), and for various values of the mean lifetime of a module 1, for a network size N=40 at time t=10000. One can see that a distributed replicated placement (detect-and-retransmit or request-schedule-then-transmit) provides the same availability as a central placement of the conflict resolution (rcCA-STAR). This can be intuitively explained since whilst the failure of a conflict resolution module in a network based on a distributed replicated placement affects only one station (1/N-th of its maximum availability), there are N modules which can fail. One can also conclude from Fig.9.3 that for a rcCA-STAR network, by introducing M-1 redundant modules in rcCA a significant improvement in network availability over networks based on a distributed replicated placement can be achieved, *even when M<<N.* In rcCA-STAR only the rcCA station needs a conflict resolution module, therefore the use of M<<N modules would give the network both cost and availability advantages over networks based distributed replicated placement where the functions associated with conflict resolution (conflict-fee scheduling, etc.) are repeated by all N stations.

The results of (2), (4) and (6) for the riskiness of the various placements are plotted in Fig.9.4 as a function of M, for varying 1, assuming N=40, and time

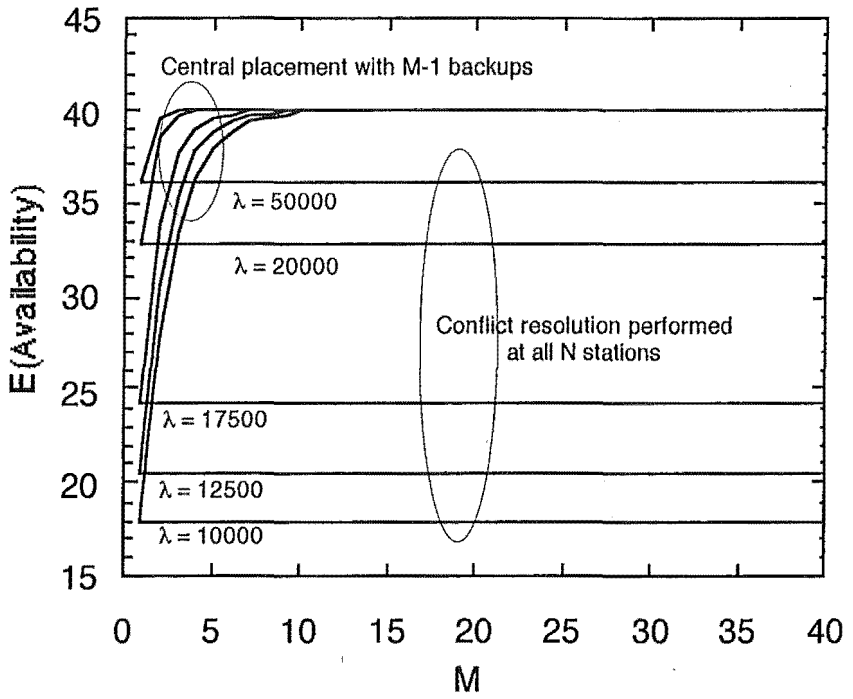


Figure 9.3: Comparison of network availability achieved by CA-Star (central placement of the conflict resolution function with M-1 backups, for M=1,2, ..., 40), with the detect-and-retransmit and request-schedule-then-transmit networks (conflict resolution repeated by all stations). N= 40 stations, T=10000.

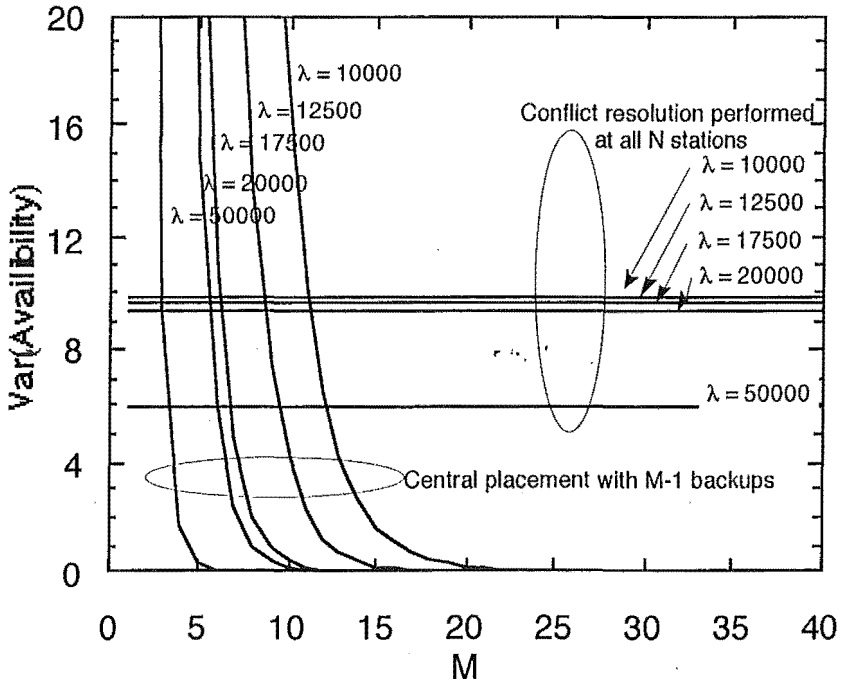


Figure 9.4: Comparison of riskiness of CA-Star (central placement of the conflict resolution function with M-1 backups, for M=1,2, ..., 40), with the detect-and-retransmit and request-schedule-then-transmit networks, N=40, T=10000.

t=10000. Define $M_{critical}(t)$ as the smallest value of M for which the rcCA-STAR network achieves lower risk than the distributed replicated placement based networks; i.e. the smallest value of M s.t. $R_3(t) \leq R_1(t)$. One can see that for λ = 10000, 12500, 17500, and 50000, $M_{critical}(10000)$= 12, 9, 7, and 4 respectively. By using M-1 redundant modules in the rcCA of a rcCA-STAR network, a significant improvement in riskiness (reduction in the uncertainty of network availability) can be achieved in addition to improved availability, *even when M<<N.*

9.8 Conclusions

Networks differ in their choice of *placement* of the destination-conflict-resolution function which specifies *where* and *when* it should be performed within a broadcast-and-select WDM star network.

In this chapter, previous placements in which the function is located at *all stations* and performed either *before* packet transmission (using the request-schedule-then-transmit concept) or after a destination conflict has been detected (using the detect-and-retransmit-if-lost concept), were compared with a central placement concept (CA-STAR), in which only one central station (CA) located at the entrance to the star coupler is responsible for detecting conflicts and (optically or electronically) buffering "otherwise lost" packets whilst they are *en route* to their destinations, until their destinations are free to receive them. The networks were compared considering the computational complexities of their MAC protocols, the complexities of the logical buffer organisation needed to support their operations, their electro-optic conversion overheads, their hardware demands, and their throughput and mean packet delay performance.

The preceding analysis showed that all CA-STAR networks enjoy substantially lower mean packet delay. This is because in the detect-and-retransmit networks, a packet experiences an additional propagation delay of at least $2a+1$ slots each time it is involved in a destination conflict (due to the time needed to detect the loss of a packet, retransmit it, and the time for the retransmitted packet to propagate from the source to destination). In the request-schedule-then-transmit networks each packet must wait for at least $2a+1$ slots prior to its transmission (for request broadcast and scheduling). In contrast, packets in CA-STAR networks can be transmitted almost as soon as they are generated, and packets involved in destination conflicts have their arrival times rescheduled to the first time-slot when their destinations are free to receive them.

The results suggests that CA-STAR networks have significantly higher throughput than the detect-and-retransmit networks (approximately 98% compared to 63%). As expected, optCA-STAR and rcCA-STAR networks also have somewhat better throughput over the "request-schedule-then-transmit" networks, since the scheduling information used by their CA is not $2a+1$ slots out of date.

Considering the time computational complexity (C_T) of the networks' MAC protocols, the optCA-FPCF/B (/R) (rcCA-FPCF/B (/R)) networks have significantly lower (C_T) than the request-schedule-then-transmit networks since under FPCF, only newly rescued packets by optCA (rcCA) need to be scheduled. Packets left in optCA (rcCA) from previous time slots are already scheduled for transmission during one of the future time slots. The worst case time computational complexity of the optCA-FPCF/B (/R) and rcCA-FPCF/B (/R) protocols were of the order O(N), compared with O(N^4) for DAS and HTDM (request-schedule-then-transmit), O(N^2) for CF-WDMA (request-schedule-then-transmit), optCA-MRS* (en route conflict resolution), and sCA/B (/R) (en route conflict resolution), and O(N) for DT-WDMA (detect-and-retransmit). The opt and rcCA-STAR networks seem especially well positioned to exploit SIMD, MIMD, or associative testing hardware for further reductions in the time needed for their computation. Due to complexity inversion, such hardware would be needed at just one station (the opt or rcCA is the only station tasked with destination conflict resolution) instead of all network stations.

The optCA-FPCF/B (/R) and rcCA-FPCF/B (/R) protocols also have the simplest logical buffer organisation requirements of the networks considered. This is due to the fact that FPCF uses the knowledge of future transmissions to streamline logical buffer organisation. Under optCA-FPCF/B (/R) (rcCA-FPCF/B (/R)), all packets in optCA (rcCA) that are scheduled for transmission during a time slot are identified by an index (E) which rotates *deterministically* after each slot. There is no need to search the buffer modules for the locations of packets that are scheduled for transmission. Thus, no queues nor lists of packets need to be created nor maintained, and no logical relationships between packets in a buffer need to be recorded.

Note that according to optCA-FPCF/B (/R) (rcCA-FPCF/B (/R)), optCA (rcCA) transmits packets that it had rescued after a delay of at most B-1 time slots. Provided that the offered load is below 98% (probability of Reflection is low), this should yield a reduction of the delay variance with respect to the "request-schedule-then-transmit" networks using the SDR, MRS or RS algorithms for conflict-free transmission scheduling. However it should

be noted this has yet to be established quantitatively.

Dividing the component count of CA (sCA, optCA, and rcCA) among stations in a network, the CA-STAR networks were found to be comparable in terms of transceiver and buffer memory demand. In particular, technologies needed for the construction of optCA (rcCA) are identical to that required for the network interface of ordinary stations. Any technological advancements that improve the data rate of ordinary stations should also enable optCA (rcCA) to match the improved rate. Also, the idea of forward planning embodied in FPCF naturally fits a simple optical implementation of optCA's (rcCA's) buffers using delay lines, since scheduling a packet for leaving optCA during the j-th future time-slot is equivalent to scheduling its delay at optCA by j time slots. Finally, most "request-schedule-then-transmit" networks require that stations are equidistant from the star coupler, so that the (conflict-free) transmission schedules computed by their stations concur. This requirement could be met, for example, by adding delay lines to their stations. CA-STAR networks, on the other hand, does not require stations to be equidistant from the star coupler, because the destination conflict function is located at just one station — the CA, (entrance to the star coupler) — which is a natural synchronisation point.

At the input to CA, data channels are SDM (on separate fibers) instead of WDM. Thus, locating the destination conflict resolution function at CA has economic advantages over providing all ordinary stations with multiple tuneable filters and delay-lines for resolving destination conflicts on multiple WDM channels. Whilst both placements employ delay lines, the central placement followed by CA-STAR obviates the need for multiple tuneable filters in the delay line implementation. The total buffer memory required is smaller since following the en route conflict resolution, packets occupy buffer "real-estate" only when necessary.

Chapter 10

Conclusions

The parallel use of multiple channels in packet switching WDM star networks mean that too many packets may simultaneously arrive for the same destination station, necessitating the implementation of a *destination-conflict-resolution* function somewhere within the network. This thesis considered explicitly, the *placement* of the destination-conflict-resolution function which specifies the location(s) *where* it should be performed, and *when* it should be performed. Traditional placements in which the function is located at *all* user stations and performed either *before* packet transmission (using the request-schedule-then-transmit principle) or *after* a destination conflict has been detected (using the detect-and-retransmit-if-lost principle), were compared with a *central placement* in which only one central station located at the entrance to the star coupler is responsible for detecting conflicts and buffering "otherwise lost" packets whilst they are *en route* to their destinations[1], until their destinations are free to receive them.

The destination conflict problem in multichannel networks was first described. Then CA-STAR architectures and protocols were proposed for implementing *en route* destination conflict resolution in WDM star networks. Thirdly, a methodology for using a network of workstations in parallel for quantitative stochastic simulation of CA-STAR networks in their steady-state was developed, and its software architecture, implementation and benchmarking discoursed (contained in a separate report). The performance, hardware demands, complexity of buffer organisation, optical-to-electronic conversion overhead, and electronic processing complexity of the CA-STAR networks were analysed and compared with WDM networks using the "detect-and-

[1]Thus the central placement of the destination conflict function is also known as "en route" conflict resolution

retransmit-if-lost" and the "request-schedule-then-transmit" principles for destination conflict resolution.

10.1 The Approach Evaluated

In the CA-STAR networks, stations may transmit packets without waiting for a long contention resolution period, and assume that each of them is successfully delivered. Thus, the minimum packet delay prior to transmission is just two slots, independent of the propagation delay. Since transmission proceeds without firstly ascertaining the transmission intentions of other stations, more than one packet may simultaneously be destined for the same destination, but the destination will be able to receive only one of them. A central arbiter (CA) sited at the entrance to the star coupler is tasked with resolving destination conflicts, freeing all other stations from the overhead, whilst avoiding functional replication. CA detects destination conflicts, buffers all packets which would otherwise be lost[2] (thereby rescuing them whilst they are *en route* to their destinations), re-scheduling their arrival times so that they reach their destinations when their destinations are free to receive them. Even if a packet is involved in a destination conflict, its delay due to propagation still equals one source-destination period. This is due to the fact that CA is located at the star coupler, through which all packets normally propagate past in their transmission from source-to-destination. This existence of a "central nexus" of physical paths may indeed be unique to the star topology.

The optCA is the simplest among the central arbiter station designs considered in this thesis. Unlike a centralised electronic switch or a station in a multihop network, optCA does not perform switching nor routing functions. optCA can be implemented using the same technologies as ordinary stations so a technology enabling a higher transceiving rate in ordinary stations would also enable optCA to match the faster rate. optCA is modularised, allowing incremental expansion to support a growing network. optCA's buffer operations proceeds at the same speed as that of ordinary stations, so they would not create an electronic bottleneck. rcCA is an adaptation of optCA that facilitates the use of simple optical buffers instead of electronic ones, thereby yielding an "all-optical" rcCA-STAR network. If rcCA is implemented using optical packet buffers, then once a station transmits a packet, it remains in the optical domain until delivered, *even if it was involved in a destination*

[2]When more than one packet simultaneously arrive for the same destination, the destination can receive only one of them. The other packets can therefore be considered as "otherwise lost" packets which needs to be rescued by CA.

conflict. rcCA-STAR networks can be implemented using half the number of channels channel (bandwidth) required by optCA-STAR networks.

10.2 Method and Scope of Analysis

The throughput/delay performance, electronic-optical conversion (E-O) overhead, hardware demand, computation complexity of the MAC protocol, and the buffer organisation complexity of several CA-STAR networks were compared to that of WDM networks using the "detect-and-retransmit-if-lost" and the "request-schedule-then-transmit" principles for destination conflict resolution.

Throughput and delay characteristics were presented in this thesis for the sCA, optCA, and rcCA based CA-STAR architectures operating according to the sCA/B, sCA/R, optCA-MRS, optCA-FPCF/B, optCA-FPCF/R, rcCA-FPCF/B, and rcCA-FPCF/R media access protocols. The results were obtained by simulating the steady-state behaviour of these networks, applying the methodology of quantitative stochastic simulation. All simulation results were produced using AKAROA, an object-oriented simulation package developed by us for automated control of precision of steady-state estimates when executing quantitative stochastic simulations in parallel time streams. With AKAROA, a sequential CA-STAR simulation program is automatically transformed into one suitable for parallel execution on multiple workstations linked by a LAN. AKAROA was also responsible for automated analysis of simulation output, using the Spectral Analysis in Parallel Time Streams method proposed in [YAU96a], which is an extension of the method of Spectral Analysis proposed in [HEID81] for uniprocessor simulations. The beginning of steady-state conditions in each simulated process was detected using a procedure given in [PAWL90]. Simulation runs were stopped when the steady-state estimates of all performance measures achieved or exceeded the required level of relative precision (typically 1% or 5%), at the 0.95 confidence level.

10.3 Main Findings

10.3.1 Performance

When a <1 (a being the source-to-hub propagation delay to packet transmission time ratio), electronic-optical conversions dominate packet delay, and it

was shown that CA-STAR has similar E/O overhead compared to the other networks. However, packets normally remain in the optical domain between source and destination. Packets are buffered by CA only if they need to wait for one or more time slots, until their destinations are free. Hence the one slot delay for O/E conversion for receiving a packet into CA's buffer (when CA is implemented using electronic instead of optical buffers) is considered desirable.

When a >1, the results showed that CA-STAR enjoys drastically lower average packet delay. This is because that when a >1, propagation delay is the dominant component of packet delay. In "detect-and-retransmit-if-lost" WDM networks, a packet experiences an added propagation delay of at least $2a$+1 slots each time it is lost (Fig. 10.1a), assuming that the source station can detect packet loss from a control channel (if acknowledgements were used instead, then each time a packet is lost, it will be delayed by at least $4a$+3 slots). In "request-schedule-then-transmit" networks each packet must wait for at least $2a$+1 slots prior to transmission, for its transmission-request to be broadcasted to other stations (Fig. 10.1b). In contrast, packets in CA-STAR networks can be transmitted almost as soon as they are generated, and would not suffer added propagation delay, even if they are involved in a destination conflict (Fig. 10.1c).

CA-STAR networks were shown to have significantly higher throughput than the "detect-and-retransmit-if-lost" networks (98% compared to 63%). CA-STAR enjoys somewhat better throughput over "request-schedule-then-transmit" networks as expected, since scheduling information used by the CA would not be $2a$+1 slots out of date.

10.3.2 Hardware Demand

As mentioned, the optCA and rcCA are the simplest among the central arbiter station designs considered in this thesis. Dividing the component count of CA among stations in a network, the optCA-STAR and rcCA-STAR networks were found to be comparable with "detect-and-retransmit-if-lost" and "request-schedule-then-transmit" networks in terms of transceiver and buffer memory and bandwidth demand, and have significantly lower hardware requirements than networks where the destination conflict problem is attacked using "receiver replication". By resolving conflicts at CA where data signals are space division multiplexed (prior to entering the star coupler), *at most one* packet per station needs rescuing per time slot. Hence, CA-STAR differs from "receiver-replication" based solutions in that neither multiple tuneable-

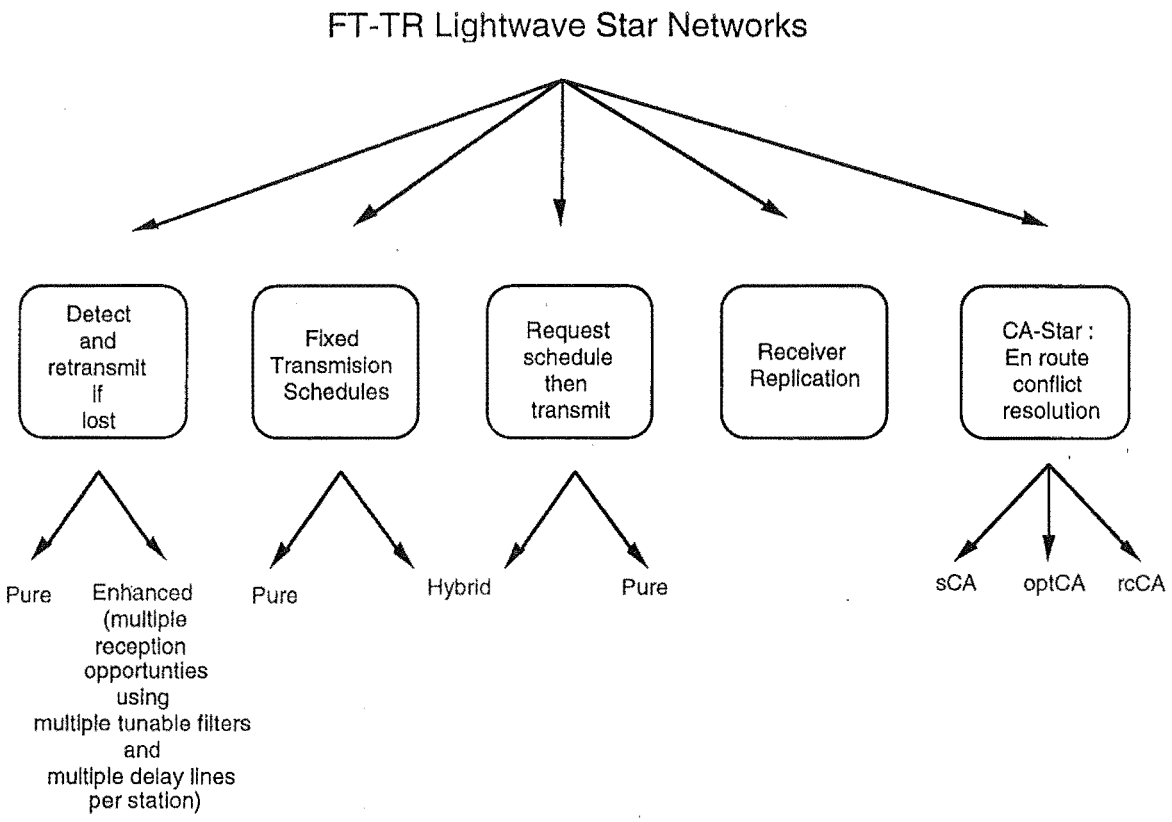


Figure 10.1: Methods for resolving destination conflicts in WDM star networks.

filters/delay-lines-and-tunable-switches nor multiple receivers are needed per station. Working with SDM signals require simpler receivers. Less buffer memory is required with the "en route conflict resolution" approach, since packets occupy buffer "real-estate" only when necessary - i.e. only while waiting until their destinations are free.

Both optCA-STAR and rcCA-STAR networks can take advantage of optical buffering for storing "otherwise lost" packets to yield a true all-optical WDM network. In an all-optical optCA-STAR or rcCA-STAR, once a packet is transmitted it remains in the optical domain until it is received by its destination, *even if it was involved in a destination conflict*. The rcCA central arbiter was shown to have the following properties :

- each buffer module of rcCA (for storing "otherwise lost" packets) is used as a packet-carrying pipe with a constant emptying rate,
- data packets remain on the wavelength on which they were originally transmitted even if they were involved in a destination conflict,

- data channels are still space division multiplexed at rcCA ;

which allow for a very simple implementation of optical rcCA buffer modules. The proposed all-optical rcCA-STAR network has these additional costs :

1. B loops of fiber are needed for a buffer module with a capacity for storing B packets.
2. either B+1 ON/OFF optical switches or one (B+1)-way optical switch is needed for a buffer module of capacity B.
3. one photonic amplifier may be required per buffer module, if ON/OFF switches are used instead of the single (B+1)-way optical switch.

The proposed all-optical rcCA-STAR network has four advantages :

1. B+1 ON/OFF optical switches are the main components required for a buffer module of capacity B.

   An ON/OFF optical switch has much reduced functionality than a tuneable optical filter (used by tuneable receivers), see section 3.2.4. In fact, a dense 2-dimensional array of light modulators is a major component of some tuneable filters. The array of ON/OFF optical switches has been used as a rapidly reconfigurable diffraction grating to provide wavelength selectivity in a tuneable filter [WARR95]. It is also the key component of some optical interconnection networks [DIAS88].

   Since even a small value of B already yields near optimal rcCA-STAR performance, only a small number (B+1) of such ON/OFF optical switches are needed per buffer module. This suggests that the cost of the ON/OFF switches of a buffer module would be small, compared to the cost of a tuneable receiver.

   An alternative would be to use a (B+1)-way optical switch in place of B+1 ON/OFF optical switches. This option would avert the power loss from splitting the input signal into B+1 parts, but it reduces the modularity of each stage.

2. Once a station transmits a packet, it will remain in the optical domain until it is received by its destination. Thus no data receivers or transmitters are needed at rcCA.
3. No (electronic) memory is needed at rcCA for storing "otherwise lost" packets.

4. At the input to rcCA, data channels are SDM (on separate fibers) instead of WDM. Thus, locating the destination conflict resolution function at rcCA has economic advantages over providing all ordinary stations with multiple tuneable filters and delay-lines for resolving destination conflicts on multiple WDM channels. Whilst both approaches employ delay lines, the central placement of the conflict resolution function followed by CA-STAR obviates the need for multiple tuneable filters in the delay line implementation. Also, since all signals are on one wavelength, if photonic amplification is required then off-the-shelf photonic amplifiers can be used. Working with SDM signals therefore bypasses the problems of gain equalisation when photonic amplification is applied to WDM channels.

Nevertheless, it should be emphasised that the optical buffering of packets was considered only to provide an alternative implementation of rcCA-STAR, not to improve its performance. The all optical rcCA-STAR alternative allows the engineer to choose an implementation based on costs and reliability considerations.

10.3.3 MAC Complexity

In optCA-MRS* networks, the optCA station used an extension of the MRS algorithm for scheduling conflict free packet transmissions from its buffers. The MRS algorithm previously used for conflict-free scheduling in "request-schedule-then-transmit" networks, was shown to have a computational complexity of the order $O(N^2)$.

The Forward Planning Conflict Free (FPCF) algorithm had been developed for reducing the computational complexity of optCA's MAC protocol. FPCF differs from previously introduced algorithms in that it plans transmissions for up to B-1 time slots in advance (B being the size of optCA's buffer modules), instead of scheduling them just for the next time-slot. This concept of forward planning naturally facilitates a simple optical implementation of CA's buffers because scheduling a packet for transmission from optCA (or rcCA) during the j-th future time-slot is equivalent to scheduling its delay at optCA by j time slots (j=1, 2, ... , B). Thus the functionality required of the buffers are much reduced. For example, in other algorithms, the duration which a packet has to be buffered is unbounded and unknown until just before its transmission. With FPCF, each packet requires buffering for at most B time-slots and the duration of stay is know prior to its entry into the buffers of

CA. Consequently, the storage functions needed for operations under FPCF can be served by a simple series of delay lines, as described in Chapter 8.

The performance of FPCF was analysed and compared the SDR algorithm, which offers the best performance from among other previously proposed algorithms. Additionally, the computational complexity of FPCF was compared with that of SDR and such approximate algorithms as MRS, K-HOL and RS, proposed for their lower computational complexity than SDR. Results showed that FPCF offers similar or even better throughput than SDR. Performance better than under SDR can be achieved, since the forward planning of packet transmissions is used to determine whether incoming packets can be transmitted conflict-free in the foreseeable future, rejecting in advance those of them for which a destination conflict within the next B-1 slots would be unavoidable.

Importantly, FPCF was shown to have drastically lower computational complexity than SDR and all other, but K-HOL, algorithms. This is due to the fact that under FPCF only packets arriving during the current time slot need to be scheduled.

Applying the FPCF algorithm for the operation of optCA and rcCA originated the optCA-FPCF/B(/R) and rcCA-FPCF/B(/R) protocols. They are among the most efficient CA-STAR protocols considered in this thesis. The "/R" versions include a deflection-routing/back-pressure procedure called "Reflection" for preventing packet loss due to buffer overflow at optCA (rcCA). Nevertheless, optCA-FPCF/R and rcCA-FPCF/R were shown to have the same or lower order of MAC computation complexity than the "detect-and-retransmit-if-lost" and "request-schedule-then-transmit" networks considered, and demands the least complex logical buffer organisation.

Networks that operate under optCA-FPCF/B (/R) and rcCA-FPCF/B(/R) seem especially well positioned to exploit SIMD, MIMD or associative testing hardware for further reductions their MAC time computation complexity. Due to complexity inversion, such hardware would be needed at just one station (the optCA is the only station tasked with destination conflict resolution) instead of all network stations.

10.3.4 Expandability

Technologies needed for optCA's construction are identical to that required for the network interface of ordinary optCA-STAR stations. Thus any technological advancements that improve the data rate of ordinary stations should

therefore also enable optCA to match the improved rate. The buffering duration of a packet at ordinary stations or at CA is independent of propagation delay, so even new stations that increase the network's diameter can be easily added without adding memory to stations' buffers.

If bandwidth use is of concern, it was shown that optCA-STAR networks can be adapted to use N instead of $2N$ channels, without notable performance degradation (rcCA-STAR).

Additional advantages of the optCA-STAR and rcCA-STAR architectures are high modularity, and incremental expandability. For example rcCA can be upgraded to serve a growing network by adding one buffer module per new station, and a fault in an rcCA module only affects the corresponding station, and it is easier to increase memory capacity of buffers by adding components at the rcCA only, instead of increasing the memory capacity of all network nodes.

10.4 Insights

Jointly, the "detect-and-retransmit-if-lost", "request-schedule-then-transmit", and "receiver-replication" strategies seem to have covered every possible angle of attack on the destination conflict problem. Not having being exposed to lightwave network literature before, I found each to be a natural solution.

Proposing an alternative therefore seemed very risky. Intuition lay behind the invention of the CA-STAR architectures. Using simulation, we have captured and quantified our intuition, comparing their performance with other WDM network architectures. Results show great improvement over networks based on the "detect-and-retransmit-if-lost" and "request-schedule-then-transmit" approaches. Moreover, the results suggests that opt and rcCA-STAR networks compares very favourably in terms of buffer memory requirement, computational complexity, and logical buffer organisation complexity. Without a standardised method for investigating new architectures, our approach was the most appropriate we know.

Still this work has several shortcomings. They fall into two categories: 1) Errors in the basis for evaluating the proposed architectures and 2) Inadequacies of the methods used for their evaluation.

1. (a) The simulation models are incomplete. For instance, how synchronisation could be achieved was not addressed, see [SEMA93]. Even if stations were synchronised, their transmissions in slots may get

out of alignment due to chromatic dispersion. Padding (i.e. gaps added to the beginning or end of each time-slot so that data packets on two wavelengths in successive slots would not overlap in time) was not considered. The timing of transmissions for each channel, so that the amount of padding required is minimised (for a given maximum network diameter) was considered in [SEMA93].

(b) Bit errors were not modelled. The bit error rate in today's fiber optic system is typically 10^{-9} or less [GREE93], far lower than that of copper wire based systems (10^{-5}). However, error correction techniques specially developed for transmission in WDM networks have already been considered. The familiar Hamming code error detection/correction scheme was applied "sideways" in [KUO94], taking the N bits of N channels during one bit duration, to generate in parallel r parity check bits. The codeword consists of the N (original) data bits, plus the r check bits. The codeword is transmitted in parallel using $N + r$ channels. When the bit error rate (BER) prior to encoding, P, is small, the coding scheme was shown to reduce the original BER to the order of P^2. When the BER is 10^{-9}, the decoded BER is about 3×10^{-17} [KUO94].

Thus bandwidth for r extra channels is traded off to keep the per channel data rate constant. This scheme suits WDM systems because the per channel bit rate is the bottleneck. Clearly, such parallel coding schemes cannot be applied to previous WDM star networks, as each station transmits data using one data channel. On the other hand, it seems that the concept can be adapted for WDM LAN/MANs.

(c) Even if the models are complete, assumptions of traffic reference patterns were used. Specifically, uniform reference was assumed in most of our simulation models.

Notwithstanding, it is common knowledge that the destination addresses of packets from each user process are non random, but exhibit a somewhat predictable pattern. This property can be refereed to as the *locality of reference*, describing the fact that over an interval of time, the addresses generated by a typical user process tend to be restricted to a subset of network stations. When the number of network-accessing user processes (multiplexed) in the station is not large, the addresses of packets generated by that station also tend to be restricted to a subset of "favourite" stations during that time interval.

Intuitively we expect the performance of WDM networks to be de-

graded under asymmetric traffic (where a fraction of all packets are destined for a subset of favourite stations). Under asymmetric traffic, the receiving capacity (one packet per time slot per station) of the few favourite stations become the limiting resource, whilst the receiving capacity of others may be idle. The simulation results for one sCA-STAR protocol under asymmetric traffic quantified this effect. Nevertheless, our study was limited to one CA-STAR architecture and protocol. We cannot answer "to what extent do the performance of the CA-STAR architectures differ *relative* to the ""detect-and-retransmit-if-lost" and "request-schedule-then-transmit" networks under various kinds of asymmetric traffic ?"

This problem has lead some researchers to evaluate new architectures and protocols in the context of specific applications (distributed databases, image transfer etc.), using trace driven simulation.

(d) Only packet switched transfer mode was considered. Ways to extend the CA-STAR protocols to support both connection oriented fixed bandwidth allocation and connection oriented dynamic bandwidth allocation and connectionless transfer modes can be envisioned. Hybrid transfer mode support was not considered.

(e) Hybrid /B-R protocols were not analysed. The simpler "/B" class of CA-STAR protocols seem better suited for most network applications assuming that the major bandwidth consumers in the network are delay sensitive but can accept some packet loss provided that the probability of packet loss is below a specified level. Using a "/R" protocol (which provide delivery guarantees) for such applications offers them a feature which is of little value, but introduces unnecessarily costs. Hybrid "/B-R" protocols, in which source stations can set a the "Delivery Guarantee" bit of packets, and where reflection (/R) is applied only if the bit is set, maybe the best compromise in the general case.

(f) Given the hardware required by the optCA-STAR or rcCA-STAR networks, we do not know a way of answering "can a better network be implemented with the same amount of resources?" We know it performs better than the "detect-and-retransmit-if-lost" and "request-schedule-then-transmit" networks, nevertheless it is important for marketers of new networks to address this question. If a competitor came up with something better with the same hardware demand, then the system could suffer from a cost/performance

disadvantage.

Given the design goals and constraints, perhaps the architecture and protocol design task can be formulated as a problem of search.

2. (a) Simulation gave us performance measures for several configurations of each CA-STAR architecture and its protocols. But, without an analytical solution, it would be expensive to explore their performance of a range of configurations.

   (b) The parallel use of network workstations yielded good speedup of simulation execution. However, a number of variance reduction techniques synergistic with SA-PTS could have been used, further increasing efficiency.

In closing, despite some shortcomings of the CA-STAR architectures and protocols, and accounting for the limitations of our method and scope of performance evaluation, the results suggests that the merits of CA-STAR (especially the optCA and rcCA variants) are sufficiently compelling for it to be worth further investigation as a candidate for WDM LAN/MAN, when low packet delay, high throughput, low MAC computational complexity, simple logical buffer organisation, insensitivity to propagation delay, and incremental expandability are of concern.

Appendix A

WDM Star Networks where Stations use Fixed Tuned Transmitters and Fixed Tuned Receivers for Data Exchange

This appendix describes FT-FR Networks and considers their main design challenges, and reviews and compares the methods used by the proposed FT-FR networks for meeting them.

FT-FR networks are characterised by use of only fixed tuned transmitters and receivers in the network interface of their stations. Each interface consists of m receivers, an $m \times m$ electronic switch, and m output buffers, and m transmitters [ACAM94], [MUKH92], [FRAT94]. Incoming packets that are received by receiver i are presented to the ith input of the $m \times m$ switch. Likewise packets from the ith output port are stored in the ith output buffer, where they wait for transmission (to the star coupler) by the ith transmitter. Typically $m < M$, where M is the number of channels used by the network.

Every transmitter (receiver) can transmit on (receive from) one and only one channel. That is, receivers and transmitters are fixed tuned to specific channels during normal network operations. The case where the transmitters/receivers can tune, but have slow tuning speed, has also been considered. In addition to the output buffers, each station may need m transmit buffers, and m receive buffers. Transmit buffers are used for storing the station's ready packets until they could be transmitted into the switch. Also, they may be used for storing transmitted packets until an acknowledgement has been received from their destination station. Receive buffers hold received packets

that are destined for the station. Up to m packets may arrive for a station during one time slot.

As $m < M$, a station cannot transmit on, nor receive from all data channels. Consequently, the packets transmitted by a station could be received by just a subset of stations in the network. Given this limitation, FT-FR networks achieve any-to-any connectivity using *multihopping*. Multi-hoping is a technique where a source station's packet is routed from its source station to its destination through multiple intermediate stations. If we represented the fact that station S_j could receive packets transmitted from S_i, by the directed edge from S_i to S_j in a connectivity graph, then such a graph constitutes the *virtual topology* of the network. The series of channels (edges) traversed by a packet in going from source to destination is often referred to as its route. An FT-FR network possesses any-to-any connectivity if every station has at least one route (that would take its packets) to every other station.

To implement multihoping, a station must analyse the destination address of any packets it receives from its input ports, deduce the appropriate output ports (based on some routing algorithm) for the received packets, and transfer the packets to the corresponding output buffer via the electronic switch. The incoming packets processed in this way will then wait in their output buffers until they could be retransmitted. Up to m packets may require processing in this manner during one time slot. Of course, if the station identifies packets destined for itself, it would switch the packet(s) to its receive buffer(s) instead.

The nature of multihop operation implies the main design decisions regarding

- virtual topology,
- choice of routing algorithm, and
- the choice of buffering, admission, and congestion control policy.

The virtual topology design problem is one of finding a virtual topology that optimises performance for a given traffic pattern. The choice of topology may be constrained to be a specific regular topological type, or irregular topologies may be allowed meaning that any connectivity pattern may be assumed. Also impacting on the choice of topology is whether nodes are restricted to executing a specific routing algorithm and buffering policy. Virtual topological optimisation is especially valuable when transmitters and/or receivers are able to be re-tuned to different channels. In this case, it may be possible to adapt the network's virtual topology to changes in the traffic pattern.

If irregular topologies are permitted, then the network's virtual topology could be closely optimised with respect to throughput or mean packet delay, given an arbitrary traffic pattern. Heuristics for finding an optimal topology assuming that irregular topologies are allowed have been analysed in [FRAT94], [BANN90a], [LABO91], [BANN90]. Irregular topologies are also more scalable in the sense that they permit any network size, up to the maximum. In comparison, many regular topologies restrict the permissible network sizes to integers of a specific sequence.

On the other hand, regular topologies permit simpler routing algorithms than irregular topologies [GANZ93]. This is possible because the low complexity of regular topologies usually admit a compact description, and has structural properties that aid in making routing decisions. However, restricting network connectivity to a regular topology constrains the solution space of the topology optimisation problem [FRAT94]. Previous work addressing the regular topology constrained optimisation and routing problems include [ELBY94], [GANZ93], [FRAT94], [ZHAN91], [KARO91], [ACAM91], [TANG94], [INES95](ShuffleNet virtual topology), [MAXE87], [FORG93], [FORG95] (Manhattan Street Network virtual topology), [TANG94a] (four-neighbour connection), and [KOVA94], [WILL93], [BANE91] (Bus or Ring virtual topologies), [BANE94] (dual unidirectional bus virtual topologies).

The main strength of FT-FR networks is that only fixed tuned transmitters and receivers are required. Single frequency transmitters and filters that meet their requirements are already available. For this reason, FT-FR networks carry the lowest technology risks of all the WDM network classes. Another strength of FT-FR networks is that they are free from the problems of packet collisions and "destination conflicts". Collisions can be averted by assigning a unique channel to each transmitter in the network.

The sacrifices for the lack of collisions and for a design which can be implemented without using tunable transmitters or receivers, are the possibility of an electronic bottleneck in the nodes, a complex station network interface, as well as a large packet delay and reduced throughput. Electronic switching and routing must be performed by all nodes. Nodes may also need to buffer some incoming packets until their outgoing link is free before retransmitting them. Consequently, nodes may also be tasked with buffering transient traffic, and with congestion control. The mean packet delay (D) is at least

$$D = H(2a + E) \tag{A.1}$$

where H is the mean number of "hops" or stations visited (counting the

destination station) by a packet in travelling from its source to its destination, a is the normalised propagation delay from stations to the star coupler (assuming that stations are equi-distant from the star coupler), and E is the average delay added to the total packet delay due to the delay at a node for packet reception, destination address decoding, making routing decisions, switching, and when waiting in the output buffer. The maximum throughput per channel per time-slot is therefore bounded by $1/H$.

Reducing H would simultaneously improve delay and throughput performance. The above mentioned topological optimisation methods represents one approach for lowering H. The transmitters and receivers of stations must be (slow) tunable, for the network's virtual topology to be reconfigurable. Also one needs to know the network traffic matrix in order to optimise its virtual topology, but the traffic pattern may vary, and the problem of forecasting it remains largely unaddressed.

In [KOVA94a] a hybrid TDM/WDM multihop network was proposed that aims to lower H by increasing network connectivity. Higher connectivity for a given topological type was achieved not by increasing m, but by allowing more than one station to transmit/receive from each WDM channel. Each WDM channel is divided into several lower capacity channels through time division multiplexing (TDM). Higher connectivity results for a given m because each fixed tuned transmitter (receiver) could transmit on (receive from) several (TDM) channels. Improved connectivity creates shorter source-to-destination routes, thereby reducing H. Notwithstanding, sharing channels through TDM reduces the number of WDM channels that could be employed by the network, lowering maximum concurrency and throughput. Also, sharing channels creates extra packet delay since a packet must wait at a station until the appropriate time slot. Therefore, an optimum degree of channel sharing exists [KOVA94a].

Another approach to reducing the average packet delay is to decrease E using deflection routing [ZHAN91], [CHAN93], [FORG95], [ACAM92], [FORG93], [ACAM92a], [ZHAN94]. Deflection routing lowers routing decision complexity and lowers the buffering delay, both of which lowers the average delay experienced by a packet at an intermediate station (E). The stations' network interface may also be somewhat simpler, and their buffer memory requirements reduced. This is an advantage because the network interface is likely to be a major cost determinant. In additional to reducing E, lower electronic processing requirements at intermediate stations mean that the electronic processing done per time slot is reduced. The duration of a time slot has to be sufficient long to allow for the electronic processing requirements at interme-

diate stations. By reducing the amount of electronic processing done per time slot, time slots of shorter duration can be used. This means that higher data rate channels can be used for a given packet size, thereby increasing network throughput. The overhead of using deflection routing in multihop networks is that packets would no longer take the shortest route to their destination [ACAM91], [ACAM92] so H is increased, especially under high traffic load.

The tradeoff between the improvement in transmission rate achievable due to reduced electronic processing at stations, versus decreased channel utilisation due to increases in H has been studied in [ACAM91], [ACAM92] by comparing hot-potato deflection routing versus minimum distance store-and-forward routing. Assuming uniform traffic, it was found that (hot-potato) deflection routing can degrade throughput to 25% of that of minimum distance store-and-forward routing, and that the relative performance of that deflection routing scheme is reduced as the number of stations grows. This implies that a significant speedup of the channel rate is needed, to compensate for the lower efficiency of hot potato deflection routing.

Appendix B

WDM Star Networks where Stations are Equipped with Tuneable Transmitters and Fixed Receivers

This appendix describes TT-FR Networks and considers the problems which have to be addressed by the media access protocols for TT-FR networks, and reviews and compares the methods used by previously proposed protocols for solving these problems.

TT-FR WDM star networks are characterised by the presence of one agile transmitter, tunable over all data channels, and one fixed tuned receiver in network interface of their stations. In addition, each station may be equipped with one or more extra fixed tuned transmitter/receiver(s) for MAC functions.

Stations exchange data as follows. Each station has its own unique data reception channel, on which all packets intended for the station would be transmitted. A source node tunes its transmitter to the channel of the destination node, and transmits according to the protocol. It is assumed that the source node could deduce which channel belongs to a destination, for instance, from the destination's address [BOGI93].

All stations listen continuously to their data reception channel, using their data receiver. After a time delay equal to one source-to-destination propagation delay period, the transmitted packet arrives to its destination (on the destination station's data reception channel), and would be received – unless the packet was destroyed by a collision.

The problem of *collisions* arises in TT-FR networks when two source stations transmit to the same destination, at the same time. Since the destination station receives data from one designated channel only, both source stations would transmit their packet on the same channel, and both packets would be lost[1].

The main strength of the TT-FR class of networks is that a packet may be exchanged between any pair of stations within the optical domain. That is once transmitted, a packet would not be delayed by optical-to-electronic conversions until it arrives at its destination– *unless it is involved in a collision with another packet.* A significant advantage of TT-FR systems compared to networks of the TT-TR class is that TT-FR networks need just one tunable component (the TT) per station [BOGI93b]. Tunable components are expected to be the major cost determinant of a station's network interface. The drawback with TT-FR networks is the problem of packet collisions.

Previously proposed TT-FR networks employed one of several techniques for dealing with collisions. One is to strictly *prevent* collisions by requiring all stations to obey a static collision-free transmission rights schedule. Another approach is to task all stations with *detecting* collisions, and with *retransmitting* their packets if lost. Lastly, collisions could be averted using special hardware together with an appropriate MAC procedure.

B.1 Collision Prevention

In the Interleaved TDMA (I-TDMA) protocol of [SIVA92], channels are time slotted and synchronised, and slots on all channels are pre-allocated for data transmission between specific source-destination pairs during every time slot. Time slots are grouped into cycles. Every station in the system has at least one chance to transmit to each other station during a cycle. Transmission permissions for all time slots can therefore be defined by a static 'allocation map' [SIVA92] which prescribes all permissible transmissions during one cycle. Packets generated by a station for transmission are stored in a FIFO transmit buffer of the network interface of the station. Once a packet reaches the front of the FIFO buffer, it will wait until the first time slot when the allocation map permits the station to transmit on the data channel belonging to the

[1] We are not aware of any proposed WDM network where the "capture" effect has been used to alleviate the problem of collisions, and in general the use of different power levels by different stations may be impractical given the already limited power budget of transmitters (power bottleneck, see Chapter 3).

destination station, and would be transmitted during that time slot. Collisions are strictly prevented by designing an allocation map which allows at most one station to transmit on the same channel during one time slot.

The I-TDMA* protocol defined in [BOGI93], [BOGI93c] improves on I-TDMA. Unlike I-TDMA, with I-TDMA* every station is equipped with N-1 transmit queues, one FIFO queue per possible destination. This eliminates the head-of-line effect present in I-TDMA, significantly improving throughput and lowering delay. Fig. B.1 contains an allocation map for a N=M station I-TDMA* network. We can see that the length of a cycle equals N-1 when there are N stations in the network. A consequence is that on average, a ready packet has to wait at least $\frac{1}{2}(N\text{-}1)$ time slots prior to transmission, even if all other stations were idle in the meantime.

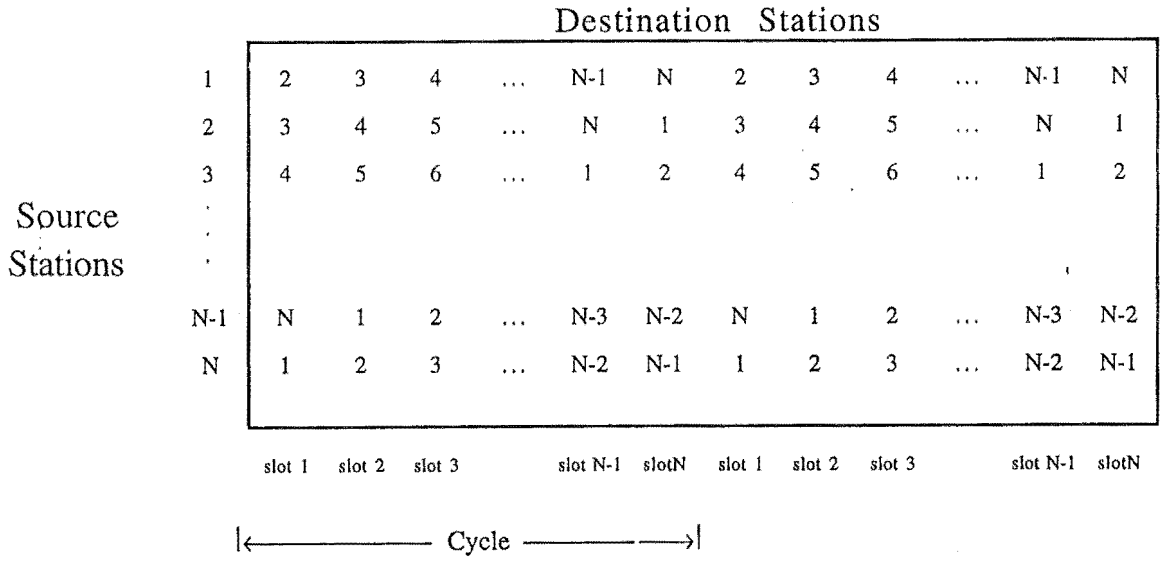


Figure B.1: Transmission permission allocation map for an I-TDMA* network with N stations and N data channels. An i,j-th element = k means that station S_i is permitted to transmit a packet to S_k during the jth time slot of each cycle.

The fixed transmission scheduling approach to dealing with collisions was also adopted by protocols proposed in [ROUS93], [ROUS95]. Four types of schedules for TT-FR systems were distinguished, and their design and optimisation for a specific input traffic matrix were analysed. First, *many-to-one* schedules were proposed which permitted more than one station to transmit to

one destination during one time slot. Collisions occur if two or more stations transmit to the same destination during a slot.

Secondly *one-to-many schedules* were proposed where a single station may transmit to more than one destination during a slot. The sets of permissible destinations of stations during a slot must be distinct (i.e. their intersections must be the null set). Thus one-to-many schedules prevents collisions. Thirdly, one-to-one schedules were studied: if S_i is allowed to transmit to S_j during a slot, then S_i may not transmit to any other station during that slot, and no other station would be allowed to transmit to S_j. *one-to-one* schedules also prevent collisions. The last type of schedule considered was *many-to-many* schedules.

Heuristics for optimising of one-to-one and many-to-many schedules were considered in [ROUS93]. Results show that, if every station always have at least one packet waiting for transmission to every possible destination (e.g.. under maximum load and uniform traffic), optimised one-to-one schedules are favoured, as no packet transmissions are wasted due to collisions. But under lower traffic, and asymmetric traffic conditions, the throughput of optimised many-to-many schedules is better than that of optimised one-to-one schedules with the same cycle length. Under such conditions, optimised one-to-one schedules have a higher probability of leaving a slot unused.

B.2 Detect and Retransmit Collided Packets

Instead of strictly preventing collisions, the Interleaved Slotted ALOHA (I-SA) protocol [DOWD91],[BOGI93], [BOGI93c] for TT-FR systems permit source stations to transmit at will. If the sender detects a collision, it will retransmit the lost packet, generally following the slotted ALOHA procedure.

For I-SA using the slot extension scheme, a time slot is divided into two phases: a data transmission phase, and an acknowledgement (ACK) phase. Stations are allowed to transmit packets during the data transmission phase, and are allowed to transmit ACKs during the ACK phase.

Hence a source station would transmit a packet during the transmission phase. If the destination station successfully receives the packet it would transmit an ACK to the source during the ACK phase. The duration of the ACK phase is composed of the time the destination needs to decode the packet header, tune its transmitter to the data channel of the source, and transmit the ACK, plus the propagation delay. The structure of a time-slot is depicted in Fig. B.2.

A feature of I-SA is that the ACK is guaranteed to be received. A source station cannot simultaneously transmit to more than one destination, so at most one ACK would be transmitted on its channel during the ACK phase. However, the maximum throughput of I-SA is bounded by that of slotted ALOHA. Also the throughput and delay performance deteriorates as the normalised propagation delay increases, because the length of the ACK phased would be lengthened too.

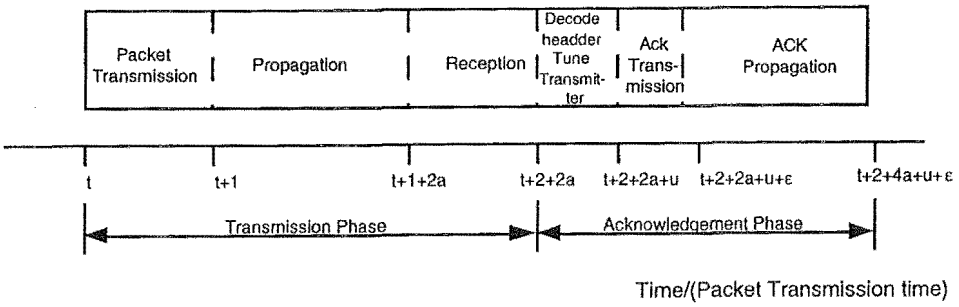


Figure B.2: Structure of a time-slot according to the Interleaved Slotted ALOHA (I-SA) protocol, where time is normalised to the packet transmission time, and the source-to-hub delay is a time slots. The time for decoding the header of a received packet and to tune the transmitter to the transmission channel is denoted by u, and the ACK transmission time is denoted by ϵ.

B.3 Random TDMA and Slotted ALOHA

In [GANZ91], every station has one tunable transmitter that is tunable over k ($1 \leq k \leq W$) of the W channels, and r receivers, $r \leq W$. Several fixed tuned receivers, each tuned to a different channel, were used to simulate a tunable receiver with limited tuning range. It was assumed that the assignment of channels to transmitters and receivers is such that single hop communication is possible. Unlike a TR, multiple fixed tuned receivers permit a station to simultaneously receive from channels corresponding to the wavelengths of its set of receivers.

A random schedule protocol, called random TDMA, was proposed for this network. All stations apply the same algorithm during every time slot to determine the set of permissible transmissions by all stations during the next slot. The algorithm constructs a collision free transmission schedule, subject

to the hardware constraints, i.e. each station could transmit at most one packet per slot, and that channel is within the tuning range of the station.

Another protocol developed in [GANZ91] is a variant of Slotted ALOHA. At the start of each time-slot, busy station S_i with a packet destined for node j transmits it with probability p_i on a chosen channel. That channel is randomly chosen from amongst the subset of channels that are both accessible by the data transmitter of S_i and accessible by a receiver of S_j. Collisions occur when two or more busy nodes attempt transmission on the same channel at the same time. The duration of a collision is one time slot. Collisions are detected, and the packets that are involved in a collision are retransmitted according to the slotted ALOHA procedure.

Results showed that the slotted ALOHA based protocol gives lower delays for low system loads, while random TDMA results in lower delays and better throughput at medium to high system loads. To determine the best choice of k and r, three 8 station, 4 channel (W=4) systems were studied: 1) k=2, r=2, 2) k=1, r=4, 3) k=4, r=1. Results for both protocols showed that network (throughput/delay) performance is best for system 2), i.e. when each station could receive from all 4 channels, and each transmitter could transmit on just one of the channels. Of the three systems, system 2) has the least number of nodes competing for each wavelength. With TDMA, this means that system 2) could offer all stations an opportunity to transmit to every other station within a lower number of time slots than the other two systems. With the slotted ALOHA, system 2) results in the lowest probability of collision. However when the number of channels is large, this design becomes infeasible due to the large number of receivers required per node.

B.4 Predict and Prevent

In another method, a station could transmit at will, but its packet is allowed to enter the star coupler only if the channel carrying the packet is not already busy. The Protection-Against-Collision (PAC) Network architecture proposed in [KARO94], [KARO91], [GLAN91a], [KARO91a], prevents packet collisions by preventing a packet from entering the star coupler, if another station is already sending a packet on the same channel.

The PAC network is made of N stations, 2 $N \times N$ star couplers, N 3-state optical switches, and N PAC circuits. Stations are interconnected by the first star coupler in the usual way. Refer to this star coupler as the *Network Star*.

Collisions are averted as follows. A station with a ready packet may trans-

mit at will, after firstly transmitting an n-bit carrier burst that precedes the packet. The access of station S_i to the primary star coupler is guarded by an optical switch, X_i. In the rest position, X_i connects the input signal of S_i to the second star coupler instead. This second star coupler was named the *control star*, see [KARO94], [KARO91a], [KARO91a], [GLAN91]. Its function is to provide the PAC circuit of S_i a multiplexed signal comprised of all the packets trying to gain access to the main Network Star, see Fig. B.3. This signal is further combined with a small fraction of the signal coming out of the Network Star. The result is a multiplexed signal comprising all the packets coming out of the Network Star plus all the carrier bursts of packets trying to gain access to the Network Star. The resulting signal is detected by a photo-detector, and its power level used to control the state of X_i (S_i's optical switch). X_i is closed only if other nodes are not transmitting on the addressed channel.

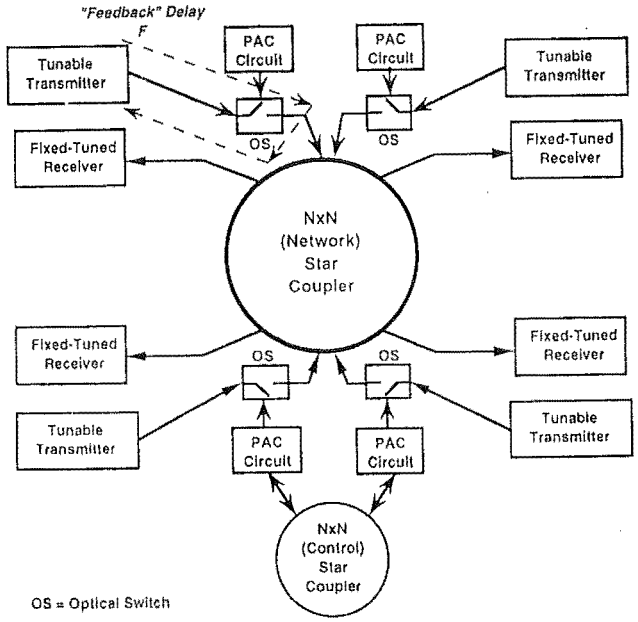


Figure B.3: Configuration of a TT-FR Network using PAC circuits, as shown in [KARO94]

Mutual exclusive access to a channel is maintained, since a station's packet is admitted to the Network star only if no other station's packet is occupying the same channel, and that once admitted, all other stations will be refrained from accessing the channel until transmission is complete. If two or more stations simultaneously attempt to access an idle channel, all their PAC circuits detect the energy of their combined carrier bursts, and all attempting stations

will be blocked.

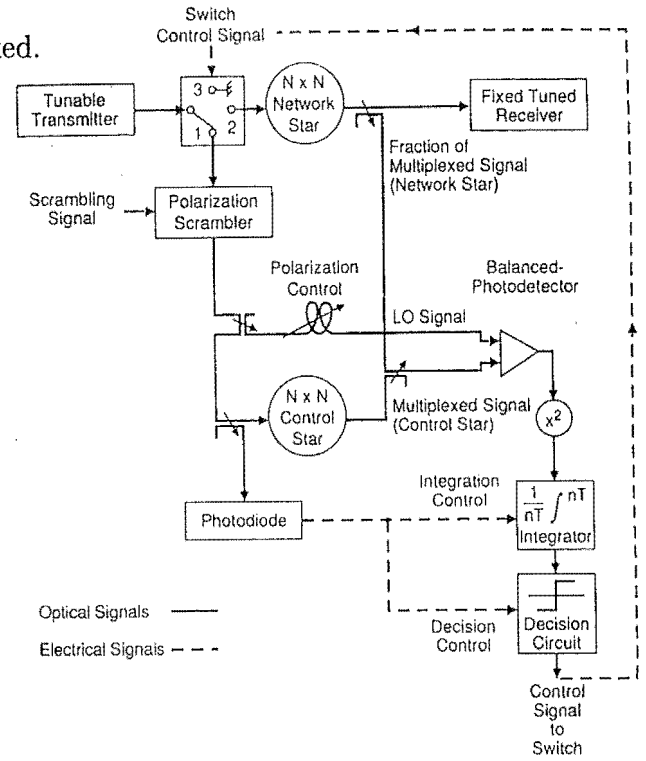


Figure B.4: Block diagram of a PAC circuit as shown in [KARO94].

If the packet of station S_i is denied access to the Network star, X_i will be set to its third position. In this position S_i's packet would be routed back to S_i. S_i could therefore detect an unsuccessful attempt to transmit by sensing for the signal of the returned packet on its transmission fibre after one round-trip-to-the-hub propagation delay period. Detection of a returned packet triggers its retransmission.

Note that S_i may pipeline its transmissions, and may also receive any packets arriving for itself during that process.

A maximum throughput of typically between 40% to 50% for uniform traffic was observed [KARO94], [KARO91]. Obviously the average packet delay is sensitive to a, since each time that a transmitted packet is blocked from entering the Network Star, its delay is increased by $2a+1$ time-slots.

Appendix C

WDM Star Networks where Each Stations is Equipped with Both Tuneable Transmitter(s) and Tuneable Receiver(s)

This appendix describes TT-FR Networks and considers the problems which have to be addressed by the media access protocols for TT-FR networks, and reviews the methods used by previously proposed protocols for solving these problems.

TT-TR networks are identified by the presence of one tunable transmitter and one tunable receiver in the network interface of their stations. Communication between stations is generally achieved as follows. A ready station selects a data channel. Then it transmits its data packet on the chosen channel following a media access protocol. After one source-to-destination propagation delay period, the intended destination tunes its receiver to the data channel, and receives the packet.

Typically, the number of data channels in a TT-TR network can be smaller than or equal to the number of stations. The ability to function using only a few data channels is an advantage when the number of WDM channels that could be accessed is limited, for example if the transmitters and/or receivers have a narrow tuning range. Like TT-FR and FT-TR systems, in TT-TR networks a packet sent by a source station remains in the optical domain until received – unless it is lost due to a collision, or destination conflict, or if the destination's receiver was not tuned to the channel carrying the packet (i.e. a lack of transmitter-receiver co-ordination).

Many protocols have been developed for resolving both the collision and transmitter-receiver problems in TT-TR systems. Nevertheless most of them ignored the problem of destination conflict, that is, the effects of destination conflicts was assumed to be insignificant in their performance analysis. Protocols for TT-TR systems can be subdivided into *random access* protocols (derived from single channel ALOHA and/or CSMA), or *fixed assignment* protocols (where the transmission rights of all stations during every slot are specified by a fixed transmission schedule).

C.1 Random Access Protocols

In many previously proposed TT-TR networks, stations wishing to communicate achieve transmitter-receiver co-ordination through the use of one or more dedicated control channels accessed according to a random access protocol. For instance, a ready station may advise its destination of the data channel to receive its packet by firstly sending a control packet containing the identification of the data channel.

Next, during the data transmission phase, the source station would send the packet on the chosen data channel, following a specific data channel access protocol. Random access protocols for TT-TR systems are therefore sometimes referred to as X/Y protocols. X is the control channel MAC protocol, and Y is the data channel MAC protocol.

Random access protocols for TT-TR networks were proposed in [HABB87], [NADE90], [GANZ91], [SUDH91a], [SUDH91b], [MUKH91], [JIAB93], [JEON95]. They assumed a network with N stations, M data channels, $M \leq N$, and one control channel.

For the protocols proposed in [HABB87], each node needs to be equipped with only one TT and one TR. No extra transmitters/receivers are needed for media access control purposes. The tuning times of the TT and TR was assumed to be zero. The N stations can transmit or receive data packets on any of the M data channels. The control channel is used by all source stations wanting to transmit a data packet, to inform the destination station of the data channel to listen to.

The X/Y protocols proposed in [HABB87] are : ALOHA/ALOHA, Slotted ALOHA/ALOHA, ALOHA/CSMA, CSMA/ALOHA, and CSMA/N-Server.

Under the ALOHA/ALOHA protocol, a busy node first transmits a control packet on the control channel and then immediately transmits a data packet

over one of the M data channels chosen at random. The control packet contains the sending station's address, destination's address, and the data channel that would be used for transmitting the data packet. Idle stations listen on the control channel. If the destination station was idle, and if it receives a control packet that contains its address, it would immediately tune to the data channel stated in the control packet (zero tuning time assumed) and then receive the data packet (see Fig. C.1).

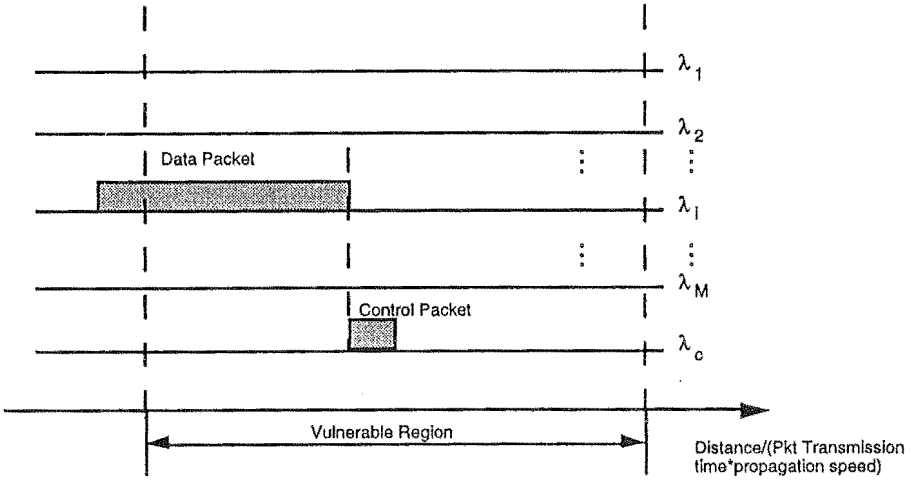


Figure C.1: Control and data packet transmissions according to the ALOHA/ALOHA protocol

Accordingly, a data packet transmission would succeed if

1. the control packet was transmitted successfully (collision free)
2. the destination was idle and therefore receives the control packet
3. no other stations transmitted on the chosen data channel during the data transmission phase

To test the validity of 1) and 3), the transmitting station listens to the broadcast of its own control and data packets respectively. If either the control or data packet is lost (due to a collision on the control or data channel), then the station must retry following the ALOHA procedure. It had been assumed

that the length of a data packet is L times that of a control packet, whose length is defined as one time unit. A collision on the control channel would occur if any other stations transmitted on the control channel during (t-1, t+1], if the station started transmitting the control packet at time t. A collision on the data channel would occur if any other station transmitted on the chosen data channel during $(t-L, t+L]$, if the station started transmitting the data packet at time t.

Notice that the validity of 2) would be destroyed if the destination was tuned to another channel (e.g.. receiving another data packet) when the control packet arrives. This situation where the control packet is lost was not considered in [HABB87].

The Slotted ALOHA/ALOHA, ALOHA/CSMA, and CSMA/ALOHA protocols are similar to ALOHA/ALOHA, except with one of the access protocols replaced by the Slotted-ALOHA or the CSMA protocol.

Under the CSMA/N-Server protocol, idle stations monitor the control channel. By monitoring the control channel over L time units, a station S_i knows which data channels would be idle, and which stations will have idle receivers (listening to the control channel). Hence if the destination's receiver would be idle, one of the data channels that would be idle is chosen, and a control packet is sent from the source station to the destination following CSMA. Thus the source first senses the control channel (on its incoming fibre). If sensed idle, transmission of the control packet proceeds. If all of the data channels were sensed to be busy, the station stops until one becomes idle.

Using CSMA/N-Server, it was assumed in [HABB87] that no collision will occur on the data channel. This is true so long as $2a<1$. Suppose S_i senses the control channel idle, and then transmits a control packet at time t. When $2a<1$, if another busy station, S_j, also transmits a control packet during (t-$2a$, t) and chose the same data channel, then the control packet of S_i would collide with the control packet of S_j. Both stations would detect the collision on the control channel and retry following CSMA. But if $2a \geq 1$, then if S_j transmitted its control packet between (t-$2a$, t-1), its control packet would not collide with that of S_i. Since both control packets would be successful, both stations will transmit their packets on the same data channel after completing control packet transmission, resulting in a collision on the data channel.

Recall that according to the Slotted-ALOHA/ALOHA protocol of [HABB87], a busy station would transmit its data packet unconditionally, after transmitting the corresponding control packet. Hence, a data packet would be transmitted even if its accompanying control packet has experienced a collision. If the control packet is involved in a collision, the packet's destination would not

know of the sender's intention to send it a data packet, nor which channel on which it should listen to, to receive the data packet. Bandwidth is therefore wasted when a station transmits a data packet when its control packet was lost.

Mehravari proposed in [MEHR90] an improved slotted ALOHA/ALOHA protocol whereby a busy station transmits its data packet if and only if the corresponding control packet was transmitted successfully. If a collision occurred on either the control or data channel, the sender waits a number of slots before retransmitting the control and data packet.

A second protocol was proposed called the slotted-ALOHA/N- server-switch protocol [MAHR90]. This protocol is similar to the above mentioned CSMA/N-Server [HABB87], except that the slotted ALOHA protocol is practised by stations for sending their control packets. If there is no collision on the control channel, the user starts transmitting the data packet on the chosen data channel immediately. Improvements in throughput performance over the CSMA/N-Server protocol was shown analytically. Also the throughput performance of the slotted-ALOHA/N-server-switch protocol is not sensitive to propagation delay.

In [SUDH91a], two sets of six slotted-ALOHA protocols and a set of reservation-ALOHA protocols were proposed. According to these protocols, packet transmission can be divided into a two phase cycle of duration $W + L$ mini-slots. W is the number of channels, and L is the data packet length (in mini-slots). The control phase is W mini-slots long, and the subsequent data phase is L mini-slots long.

In two of the protocols, the W mini-slots in the cycle are pre-assigned to the data channels. A busy station wanting to transmit a data packet on λ_i must firstly transmit the corresponding control packet on mini-slot i during the control phase. The control packet could be lost due to a collision with another control packet transmitted on mini-slot i. The fixed assignment of control slots to data channels ensures that if a control packet is successful, then the corresponding data packet will also be successful. In the first protocol, all stations execute the control phase during the first W mini-slots, then all enter their data phase during the next L mini-slots, and so on. In protocol two, the control phase of one cycle occurs during the data phase of the previous cycle. The duration of each phase is therefore equal, and is max(W,L). Protocols three to six are variants of protocol two.

The second set of slotted ALOHA protocols presented in [SUDH91a] parallels the above, except that a station transmits the data packet signalled during the control phase, only if the corresponding control packet is successful.

Two reservation protocols were also defined in [SUDH91a]. These were targeted for circuit switched traffic.

Notice that data channels would always be idle (wasted) during the control phase of every cycle, under the first protocol of each set in [SUDH91a]. Likewise the control channel would be idle during the data phase. Multi-control channel slotted Aloha protocols were proposed to take advantage of this otherwise unused bandwidth in [Sudh91b]. Also methods for interconnecting WDM star networks using variations of the Aloha protocols were discussed.

It had been highlighted in [MUKH91] that the throughput performance of many X/Y protocols exhibit a bi-modal behaviour, when considered as a function of offered load. The reason is that if the number of data channels is small, data channels become the bottleneck. As the number of data channels is increases, the control traffic also increases, since the probability that an idle data channel is found would also increase. Consequently, when the number of data channels become sufficiently large, the control channel would become the bottleneck. Also, successfully transmitted control packets may be lost, if the receiver of the intended destination station was tuned to a data channel. The effect of this 'receiver collision' problem was also studied in [MUKH91] and shown to be more significant for small user populations.

To summarise, the X/Y random access protocols suits stochastic traffic well under light traffic load. Also many of them do not require dedicated transmitters nor receivers for MAC purposes.

Notwithstanding, the performance of X/Y protocols deteriorate under high load. Another drawback of all above random access protocols is that their delay performance still degrades with increasing propagation delay, since packets may be lost. Each retransmission of a packet would add at least $2a$ to the packet's total delay. Moreover, data packets are delayed by at least $2a$ slots prior to transmission, even if all data channels were idle, using the protocols that require the outcome of the control packet to be known before initiating data packet transmission. Another problem with some of then is that their throughput deteriorates with increasing normalised propagation delay (those employing CSMA), or are limited to a maximum of 18% (ALOHA based protocols).

C.2 Collision Free Transmission Schedules

Fixed transmission schedule based protocols were also developed for TT-TR systems [GANZ92], [PIER94], [ROUS93], [GANZ94]. These protocols are

similar in principle to those discussed for TT-FR systems, see section B on page 237, except they are extended to take advantage of the tunability of the data receiver, and to accommodate the case where $W < N$. Also, several of them account for no-zero transmitter and receiver tuning times when searching for transmission schedules that provide any-to-any connectivity.

Appendix D

Influence of Modelling Assumptions

In this Appendix, we study the influence of the assumptions introduced in the standard model that was used for performance analysis throughout this thesis (the standard model was first described in section 4.2.7). The influence of the assumptions were investigated taking the sCA/R network as a benchmark. First the impact of the distribution of the destinations of packets was studied, by considering a non-uniform reference model. This model is used to test the performance of the networks in situations where stations can be divided into logical work groups, where the level of intra group traffic differs from the level of traffic between groups. In both the standard model and the non-uniform reference model, traffic is symmetrically distributed in the sence that the mean packet arrival rate to stations are the same for all stations. As a second step, the assumption of symmetric load distribution is relaxed by assuming that a subset of stations is more likely to be the destinations of packets. Thridly, the Bernouii new packet generation process assumption is relaxed by assuming that the new packet generation process of each new station is defined by a Markov Modulated Bernouii Process model.

D.1 Non-Uniform Reference Model

In the standard model, packets originating at a station were addressed to any of the other (N-1) stations with equal probability. This uniform reference assumption was considered to facilitate comparison of results with those obtained in most previous studies. Notwithstanding, in many applications

destinations of packets generated by a station maybe not purely random, but can be restricted to subsets of network stations. This property is refereed to as the *locality of reference* .

Let N stations be partitioned into N/G mutually exclusive groups. The group to which a station belongs will be called its *local group*, the other groups being *remote groups* with respect to that station. We will investigate sCA-STAR networks where new packets generated by a station are destined for a station in its local group with probability $h/(G\text{-}1)$, and are destined for one of the other stations (in any remote group) with probability $(1\text{-}h)/(N\text{-}G)$. Thus, the probability that a packet would be destined for any station in the origin station's local group is h, called the local group *hit ratio* hereafter.

Fig. D.1 shows the throughput in a sCA-STAR network with N=10 stations for two levels of traffic, when B is very small (B=2.4). Every group has G=2 stations, giving 5 groups in total. We observe from Fig.11 that under high traffic, throughput *improves* as h increases. At h=1, throughput equals unity, as expected. The reason is that when G=2, all local group references would be to the same station in its local group (a station would not transmit to itself). Therefore when h=1, there would be no destination conflict, hence throughput reaches its maximum. When p=0.10, throughput is already maximised, thus as shown by the graph for p=0.1 in Fig. D.1, throughput equals approximately 0.1, irrespective of h.

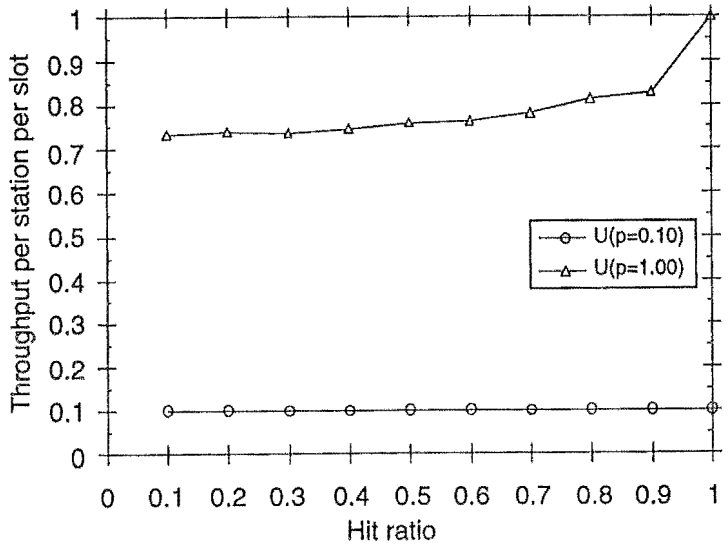


Figure D.1: Throughput of sCA/R vs. h, for p=0.1, and p=1.0, N=10, a=5 slots, B=2.4, G=2. Relative precision $\leq$5%

Fig. D.2 contains the average packet delay for the same network. When p=1, packet delay decreases from approximately 16 slots, to 11 slots, as h increases from 0.1 to 1. Under low traffic (p=0.1) average packet delay does not change notably with increased h.

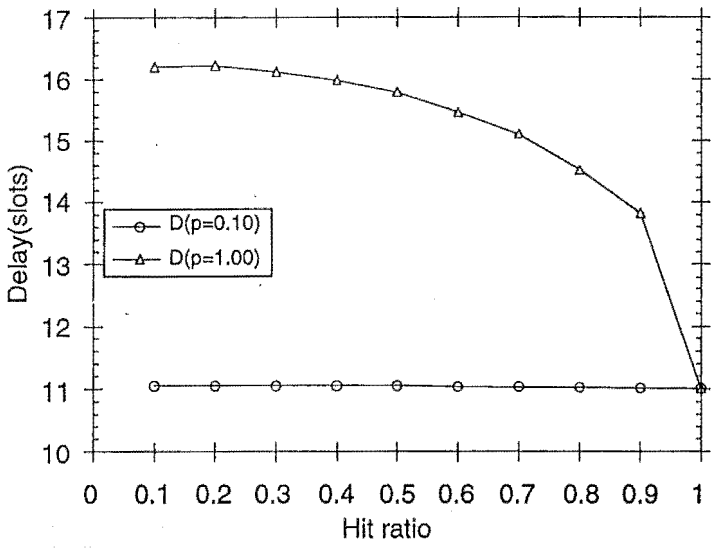


Figure D.2: Packet Delay of sCA/R vs. h, for p=0.10 and p=1.0. N=10, a=5 slots, and B=2.4, G=2. Relative precision $\leq$5%

Next results for a sCA-STAR with N=20 stations are shown in Figs. D.3 and D.4. Stations are partitioned into $20/G$ groups where G=4, hence there are 5 groups. Buffer size is B=2.2, hence just 4 packets can be kept in central buffer, (tempbuff's capacity is 40 packets). We see that Figs. D.3 and D.4 reaffirms the relationships between throughput/delay and h observed in the N=10 station networks. Also results suggests that increasing network size lowers the impact of changes in h.

The general conclusion is that increases of h (localities of reference) unambiguously improves performance. Performance improvements are more significant under high traffic, since network performance is already close to ideal when traffic is low.

D.2 Asymmetric Reference Model

The third traffic model we analysed represents a worst case traffic scenario for sCA-STAR (and many single hop WDM networks in general). All stations

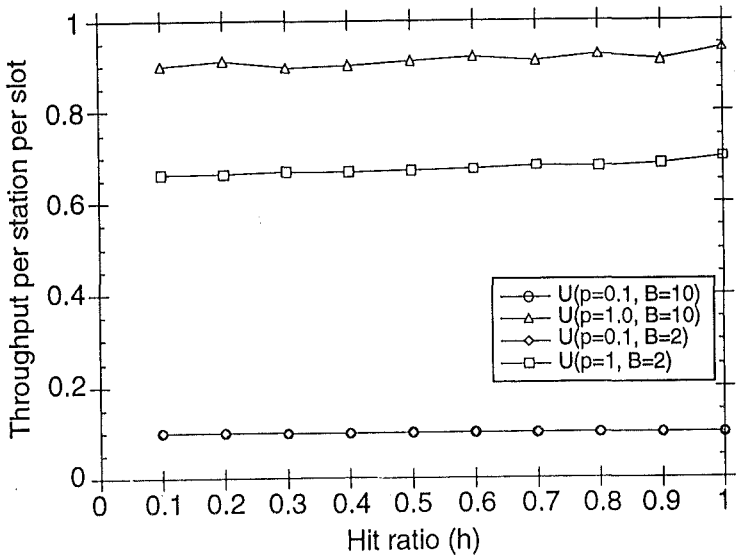


Figure D.3: Throughput of sCA/R vs. h, for p=0.1, and p=1.0, N=20, a=5 slots, G=4. Relative precision $\leq$5%

generate packets destined for a fixed subset of G stations with probability h, and destined for any one of the other possible N-G destinations with probability (1-h). Let the first subset consisting of G stations be called the *common group*, and the N-G other stations be called the *global group*. Packets destined for a group would choose one of the possible destinations within the group with equal probability. When h=1, then all network traffic would be destined for the common group. This we name the Asymmetric Reference Model (ARM), representing situations that may occur, for example, in distributed computation (e.g. a group of slave stations generating results for processing by a smaller group of masters), process control and monitoring (group of stations sending observations to a controller), and distributed databases (active user(s) viewing relations joined from several base relations stored on other servers). For a toughest test, we focus on the worst case where all stations generate a packet with probability one in every slot (i.e. p=1).

The first question such ARM scenarios raise is "Is the sCA-STAR network stable ?" When $G << (N\text{-}G)$ a large number of stations would be transmitting to a small subset, and the network's usable receiving capacity is only a fraction of the transmission rate. The throughput and mean delay results for a 20 station sCA-STAR with ARM traffic input are summarised in Figs. D.5 and D.6 respectively. A common group size of G=4 stations was assumed, so when h=1, all network traffic would be destined to just 20% of stations. An examination of Fig. D.5 shows that sCA-STAR provides the optimum throughput

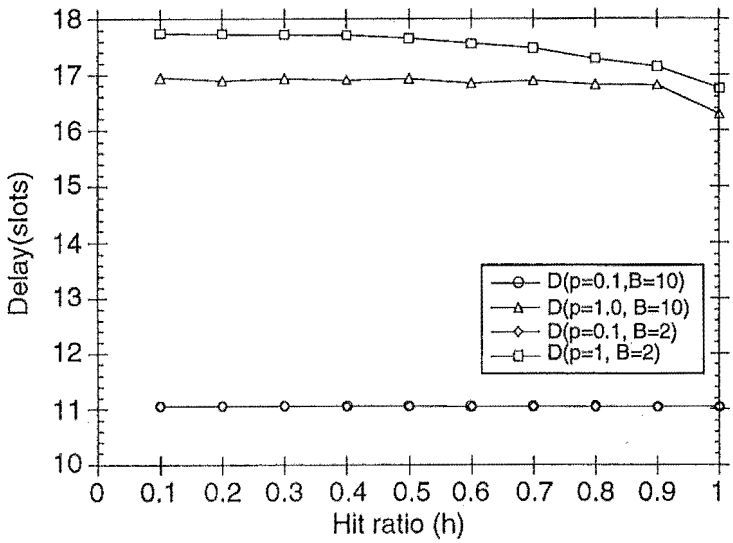


Figure D.4: Packet Delay of sCA/R vs. h, for p=0.10 and p=1.0. N=20, a=5 slots, and G=4. Relative precision $\leq 5\%$

in the worst scenario. When h=1 the maximum packet throughput that could be supported by the network, given the constraint of one data receiver per station, is 0.2, and Fig. D.5 demonstrates that sCA-STAR achieves that for all buffer sizes studied. As expected throughput is maximised when h=0.2. An examination of Fig. D.6 shows that sCA-STAR is indeed stable, the average packet delay being bounded in all cases. Yet, when p=1, the new packet generation rate is 20 packets per slot, but when also h=1, the maximum reception rate is just 4 per slot. To explain stability in face of this mismatch, we must consider the reflection mechanism. Reflection acts as an *input regulator*, in addition to its obvious role in eliminating packet loss. According to the sCA-STAR MAC protocol, whenever a station has a reflected packet buffered it will block the submission of new generated packets from its LLC layer until the reflected packet is transmitted. With just 3 packet buffers per station this is necessary to ensure that a station always has at least one free buffer, hence all reflected packets to itself can be received and retransmitted (just one free buffer is needed since a station would receive at most one reflected packet per slot, and can also transmit one packet per slot). Thus, reflection regulates the input of new packets into the network. The results show that reflection acts as an ideal regulator, yielding a throughput that equals reception capacity.

We also see that delay increases when sCA buffer size ($20B$) increases. Since throughput is the same for all values of B when h=1, p=1, an interesting question is whether increasing B worsens overall delay performance then ? To

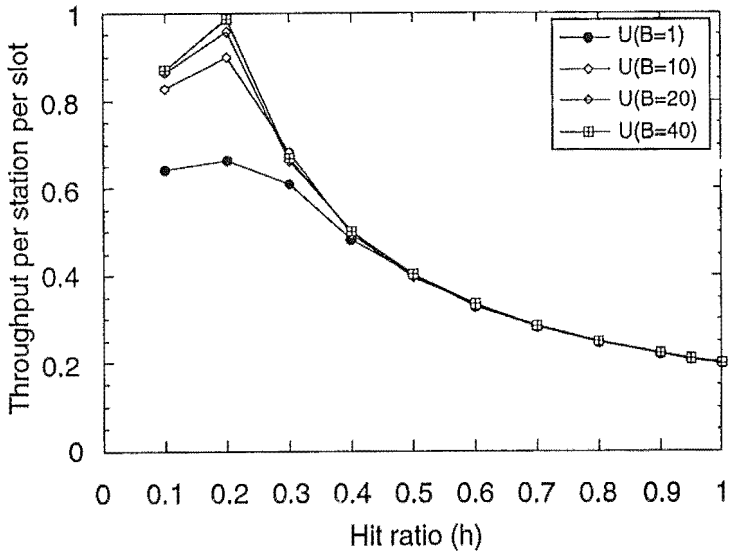


Figure D.5: Throughput of sCA/R vs. h, when p=1, N=20, a=5 slots, and G=4. ARM. Relative precision $\leq$5%

answer this question, let us consider the dynamics of the sCA-STAR when h=p=1. sCA would almost always be full, since reception capacity is 20% of transmission rate. One consequence is that sCA can be viewed as part of a pipeline from source to destination. More buffers for sCA merely lengthens the pipeline, but could not increase the network's reception rate. Hence MAC delay increases with increased B. However, a lengthened pipeline means that more packets could be admitted to the MAC layer, instead of being blocked (rejected). Hence we expect that *end-to-end* delay experienced by applications would be independent of B, even in this worst case scenario.

D.3 Markov Modulated Traffic Source Model

Common to the three previously analysed models is the Bernoulli new packet generation process assumption. Now, to evaluate the validity of this assumption we consider the more general case where the new packet arrival process to each station is a superposition of independent data traffic sources, as well as bursty sources such as multi-packet data transfer (e.g. file or image transfer), and packetized voice. In addition to correlated arrivals, each packet burst is likely to have the same station as its destination. For example, a series of packets corresponding to a file to be transmitted to a station would all have that station as their destination. We represented such a multiplexed mixed traffic

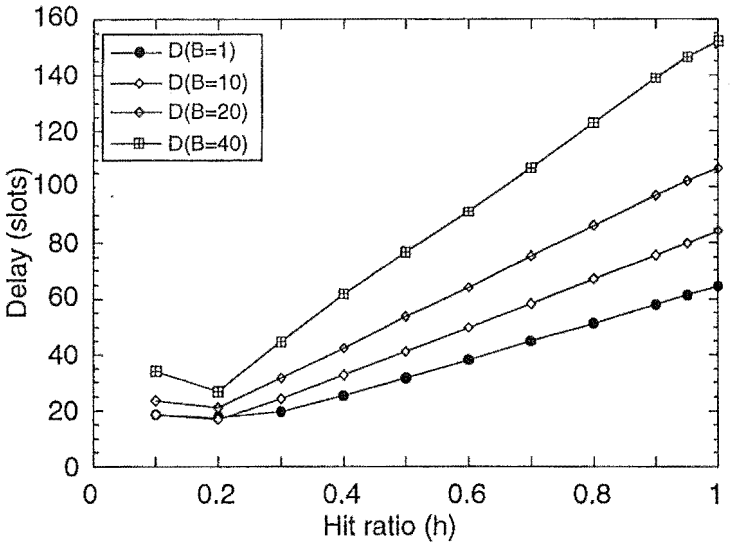


Figure D.6: Packet Delay of sCA/R vs. h, when p=1, N=20, a=5 slots, and G=4. ARM. Relative precision $\leq$5%

source using a Markov Modulated Geometric Process (MMGP). A MMGP is a doubly stochastic Bernoulli process where the rate is determined by the state of a discrete time Markov chain. We use a two state Markov chain where the mean sojourn times in states 0 and 1 are $1/\lambda_0$ and $1/\lambda_1$, respectively, where λ_0 and λ_1 are the probability that a station would depart from states 0 and 1 respectively during a slot. When the chain is in state 0 the new packet generation process is Bernoulli with rate p_0, where each packet generated when the station is in this state would choose a possible destination randomly with equal probability. Each time the chain enters state 1, a destination would be chosen randomly from the set of possible destination stations. All packets generated would have that station as their destination until the station departs from state 1. In state 1 packet generation is also governed by a Bernoulli process, but with rate p_1. A new series of sCA-STAR simulations were conducted using this MMGP source model for B=2.4, and B=10, and for ($1/\lambda_0$, $1/\lambda_1$) = (140, 60) and (100, 100) respectively. Two levels of average offered traffic, for p_{tot}= 0.5 and 0.8, were considered, where $p_{tot} = p_0\ (\lambda_1\ /(\lambda_1 + \lambda_0\ )$ $+\ p_1\ (\lambda_0\ /(\lambda_1 + \lambda_0\ )$.

Throughput and delay estimates from this analysis when B=2.4 are presented in Fig. D.7 and D.8 respectively, as a function of p_1. An examination of Figs.D.7 and D.8 shows that at both traffic levels, and for both pairs of ($1/\lambda_0$, $1/\lambda_1$), performance degrades as p_1 increases. This suggests that we should expect performance to degrade under mixed traffic, for B very small. Notice that

when p_{tot}=p_0=p_1 the model reduces to the standard Bernoulli source model except that when a station is in state 1, then all packets it generates will be sent to the same destination chosen on arrival to that state. Comparing results for this case with those of the standard model for the same sCA-STAR network, we find that such "bursty destination selection" also contribute to lower performance. However we found that when B=10, the sCA-STAR networks for the same traffic levels show little degradation from the standard source model. This suggests that sCA-STAR performs well under mixed traffic too, unless sCA has insufficient buffers. Here at least B=10 is recommended.

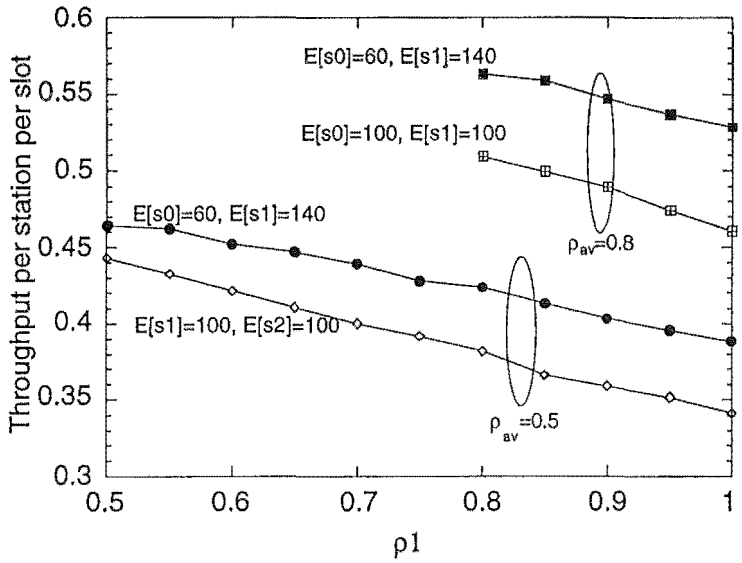


Figure D.7: Throughput of sCA/R vs. p_1, for p_{tot}=0.5 and 0.8, and $(1/\lambda_0,\ 1/\lambda_1)$ = (140, 60) and (100, 100). N=20, a=5 slots, B=2.4, and G=4. Relative precision $\leq$5%

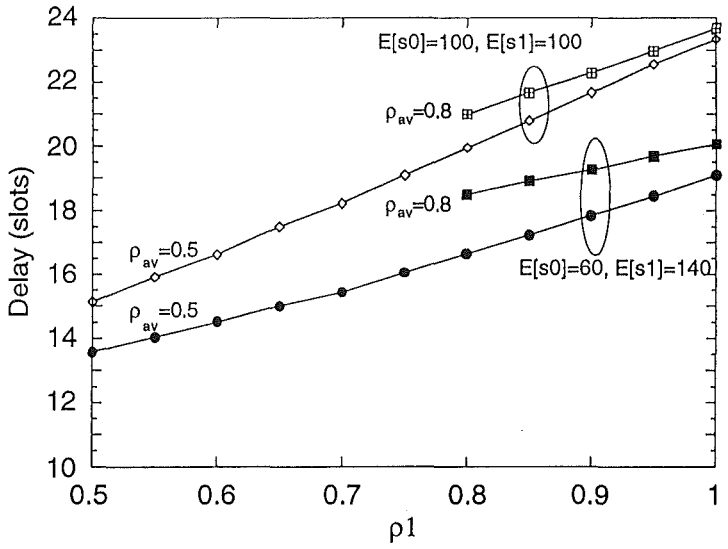


Figure D.8: Packet Delay of sCA/R vs. p_1, for p_{tot}=0.5 and 0.8, and $(1/\lambda_0,\ 1/\lambda_1)$ = (140, 60) and (100, 100). N=20, a=5 slots, B=2.4, and G=4. Relative precision $\leq$5%

Appendix E

Definition of the sCA/R Protocol

This Appendix provides a specification of the sCA/R protocol using pseudo-code.

MAC Protocol for Ordinary Stations

The sCA/R protocol is a variation of the sCA/B protocol. Its packet Transmission procedure and its Arbitration(H,R) function for ordinary stations is the same as that of the sCA/B, see section 4.2.3. The Reception procedure is a modification of the Reception procedure for ordinary stations in sCA/B. According to sCA/R, S_i listens on λ_{N+i} if it was not expecting to receive a packet destined for itself. If it receives a reflected packet on λ_{N+i}, then it transfers that packet to its transmit buffer, and blocks the arrival of a new packet (if any) during the next time-slot.

Let S_i (i=1, 2, ..., N) maintain the variables H, N, R, and I, which are as defined earlier in the case of sCA/B. Let Z be a flag such that at the beginning of t, Z==0 if the station did not receive a reflected packet during $t-1$; Z==1 o.w.

Procedure Station Reception (Executed by S_i (i=1, 2, ..., N) during every slot)
Begin
CoBegin
if (Z ==1) **then**
{ Z =0 ; **block** the arrival of any *new* packet from its LLC layer;
transfer reflected packet (received during previous slot) to its transmit buffer ; }

```
        { receive the addresses in the mini-slots and assign their contents to H;
        receive the recall field and assign its contents to R; }
        if (I ≠ 0) then receive the packet from λ_I ;
          else { receive packet from λ_{N+i}, if any ;
            if a packet was received then Z=1 ; }
    CoEnd ;
    I = Arbitration(H,R) ; // As specified in sCA/B
End ;
```

Note :

The Z=0 assignment would be done before the (conditional) execution of Z=1.

MAC Protocol of sCA

sCA receives and buffers only packets that would otherwise be lost, transmitting them to their destinations when appropriate. In addition, if its central buffer is full and there are packets which should, but cannot, be accommodated, then sCA diverts (transmits) them to surrogate destination stations, following the Reflection procedure.

Let randomi(m, n, r=rand()) and rand() be the uniform integer and uniform real random number functions, respectively. L_0 denotes the central buffer capacity, and L_i (i=1, 2, ..., N) denote the length of each of the ith queue in the central buffer as defined in section 4.2.5 (note that the value of L_i includes the packet in the head of the queue which is being transmitted during the current slot, if any). As before, we follow conventional notation by defining B to be the buffer memory capacity per station. In CA-STAR networks buffer memory is needed mainly by sCA instead of by all stations[1], since ordinary stations can send packets almost without delay, and can assume their successful delivery once transmitted. Hence, we assume that in an N station network, the memory of B packets per station is located at the sCA, instead of being distributed amongst stations. Let the tempbuff buffer (logical) region be allocated a capacity for storing $2N$ packets. As before, let NB_{cb} be the capacity of the central buffer region of sCA. Thus, in the case of sCA/R networks,

$$B_{cb} = B - 2 \tag{E.1}$$

and

[1]strictly, each station needs a transmit buffer capable of storing up to 3 packets

$$L_0 = (B-2)N \tag{E.2}$$

Let mod(x,y) be the modulus function. Accordingly,

$$mod(x,y) = x - (y\lfloor x/y \rfloor) \tag{E.3}$$

The sCA station maintains the variables H, R, P, and Y as in the sCA/B protocol. Let :

- U=$[u_i]_{N\times 1}$ be a column vector representing the $2N$ locations of tempbuff. At the beginning of a time-slot, u_i=1 if the ith location of tempbuff contains a packet; u_i=0 o.w.. At the beginning of a time-slot, the occupancy of tempbuff equals the value of Y, thus u_1+u_2+ ... +u_{2N}=Y.
- M =$[m_i]_{N\times 1}$ be the planned Reflection matrix, and
- G =$[g_i]_{N\times 1}$, $i \in 1, ..., N$ be the tempbuff-to-central-buffer-transfer matrix.

At the beginning of t, $m_i > 0$ if during t sCA should transmit (reflect) the packet from u_{m_i} to S_i. Hence S_i would act as the surrogate destination for the packet in u_{m_i}. At the beginning of t, $g_i > 0$ if during t sCA should "transfer" the packet from u_{g_i} into the central buffer.

Procedure sCA Transmission (Executed by sCA during every slot)
CoBegin
transmit R on the current recall field using T_c;
forall T_i $i = 1, 2, ..., N$ **doparallel**
if ($L(i) \geq 1$) **then** { **transmit** the packet from the front of Q_i on λ_{N+i} using T_i ; Y=Y - 1 ; }
else /* Reflect packet */
if ($m_i > 0$) **then transmit** u_{m_i} using T_i ;
CoEnd

The Reception Procedure of sCA According to sCA/R

By monitoring mini-slots during t, sCA can identify destination conflicts and hence deduce which incoming data packet(s) need to be received during t+1.

If the packet on channel λ_i needs to be received, then sCA sets p_i to 1. In addition, during t sCA also plans which packets in tempbuff should be transferred into its central buffer (recording them in array G), and which (if any) needs to be reflected (recording them in array M). The Reception procedure is specified as follows.

Procedure sCA Reception /* Executed by sCA during every slot, t=1,2, ... */
Begin
  CoBegin
    receive mini-slots and assign the addresses therein to H;
    for all receivers R_i i=1, 2, ..., N **doparallel**
      {
       if (p_i == 1) **then** receive the packet from λ_i using R_i ;
       if ($g_i > 0$) **then transfer** u_{g_i} into the central buffer ;
      }
  CoEnd
  P = Plan_Receptions(H) ; // Update P, M, and G
End;

Procedure Plan_Receptions(H) /* executed during "tuning period" */
  (register) integer j, k ;
  integer v, w, $spos$;
  conflict analysis matrix, $D = [d_{i,j}]_{N \times N}$, $d_{i,j} \in 0, 1, ..., N$;
  conflict count matrix, $U = [u_i]_{N \times 1}$, $u_i \in 0, 1, ..., N$;
  surrogate candidate matrix, $S = [S_i]_{N \times 1}$, $s_i \in 0, 1, ..., N$;
Begin
  v = rand() ;
  $[u_1, u_2, ..., u_N] = [0, 0, ... , 0]$;
  $[p_1, p_2, ..., p_N] = [0, 0, ... , 0]$;
  $[r_1, r_2, ..., r_N] = [0, 0, ... , 0]$;
  spos = 0 ;
  for j=1 to N do
    if ($h_j > 0$) **then**
      { $u_{h_j} = u_{h_j} + 1$; $d_{h_j, u_{h_j}} = j$; }
  for j=1 to N **do**
    { **if** ($u_j > 0$) **then**
      { **if** (L(j) >1) **then** { $w = 0$; $r_j = 1$; }
          else w= randomi(1, u_j, v) ; }
      for k=1 to u_j **do**
          if ($k \neq w$) **then** { $p_{d_{j,k}}$=1 ; T=T+1 ; }
    }

```
    /* Next, find stations which may serve as surrogates */
    for  k=1 to N do
        if (u_k == 0 and L(k) ==0) then { spos=spos+1 ; s_spos=k ; }
    /* Plan transfers of packets into central buffer, and the reflections of packets if needed *
    Plan_Reflection(S, spos) ;
    return (P) ;
End ;


Procedure Plan_Reflection(N × 1 array S, integer spos) /* executed during "tuning period
   integer j, k, b, r, c ;
   integer num_assigned, free_space, to_reflect, to_transmit ;
Begin
    for j=1 to N do
       if (L_j > 1) then to_transmit=to_transmit+1 ;
    free_space = L_0- (Y - to_transmit) ;
    /* note: Now T equals tempbuff occupancy, plus the */
    /* number of packets to receive during the next slot */
    to_reflect = max{0, (T - free_space - N)} ;
    k=randomi(1, 2N) ; r=0; c=0 ; num_assigned=0 ;
    // Select packets for transfer or reflection from tempbuff
    for j=1 to 2N do
        if (c < free_space and u_k == 1) then { c=c+1 ; g_c = k ; k= mod(k+1, 2N) } ;
          else
            if (num_assigned < to_reflect and u_k == 1) then
              { num_assigned=num_assigned+1;
                m_{S_num_assigned}=k ; k=mod(k+1, 2N) ; }
End ;
```

Appendix F

Computational Complexity Analysis

This Appendix derives the computational complexities (C_T) of CA-STAR protocols (en route conflict resolution), as well as the CF-WDMA (request-schedule-then-transmit) [CHEN91], [CHEN92] DT-WDMA (detect-and-retransmit) [CHEN90], [PAPA92], DAS (request-schedule-then-transmit), and HTDM (hybrid request-schedule-then-transmit/fixed transmission schedule) [CHIP93] protocols.

All variables referred to in the following analysis of CA-STAR protocols are as declared in the definition of the respective protocols (recall that the scope of variables declared in the definition of a CA-STAR protocol is limited to that protocol).

F.1 Computational complexity of the sCA/B protocol

In this section we give expressions for the time computational complexity and the network computational complexity of the sCA/B protocol.

CA-STAR differs from the other networks in that the CA station's MAC has higher computational complexity than ordinary stations. The $C_T(sCA/B)$ therefore equals the time complexity of the MAC protocol of the sCA station.

During one time-slot, the Transmission Procedure of sCA need N comparisons (to test if queue i, i=1,2, ...,N is empty), and up to N subtractions and N assignments (to decrement Y). Consequently,

$$C_T(Transmission\ Procedure\ of\ sCA/B) = 3N \qquad \text{(F.1)}$$

The Reception Procedure of sCA/B makes one call to procedure Plan_Receptions(H) during each time slot. The Plan_Receptions(H) procedure requires at most N assignments for initialising each of U, P, and R to zero.

Its first j-for-loop requires at most N comparisons, $2N$ assignments, and N additions to generate U (where u_i equals the number of packets destined for S_i).

Each iteration of the second j-for-loop requires at most 1 comparison, 1 multiplication and N subtractions (for executing randomi(1, u_j, v)), and N comparisons, $2N$ assignments and N additions (for the inner k-for-loop). The second j-for-loop has N iterations, thus it requires $N(2+5N)$ scalar operations.

The third j-for-loop generates R. It has N iterations, each of which requires at most 2 comparisons, 2 assignments and 1 addition operation. Hence the third j-for-loop requires at most $5N$ scalar operations. Adding, we find

$$C_T(Plan\_Receptions(H)\ of\ sCA/B) = 11N + 5N^2 \qquad \text{(F.2)}$$

The Reception Procedure requires N comparisons (testing if $p_i == 1$); one subtraction, comparison and assignment to execute j=min(L(0)-Y, T). There are $0 \leq T \leq N$ packets in tempbuff. j, $0 \leq j \leq T$, of them are randomly chosen (for transfer into tempbuff). We propose the following procedure for randomly selecting j packets from T without replacement.

Procedure Random_Selection(integer j, T)
Variables
integer u, i, k ;
S =$[s_i]_{N\times 1}$, $i \in 0, 1, ..., N$;
R =$[r_i]_{N\times 1}$, $i \in 0, 1, ..., N$;
B =$[b_i]_{N\times 1}$, $i \in 0, 1, ..., N$;
Begin
$u = T$;
for $i = 1$ to T **do**
b_i = i ;
for $i = 1$ to j **do**
{ r_i = generate a random number in [1,u] ;
u = u-1 ; }

for $i = 1$ to j **do**
{ $selection_i = b_{r_i}$
for $k = r_i{+}1$ to T **do**
$b_i = b_{i+1}$; }
End ;

We assume as in [CHEN94] that the algorithm for generating a (pseudo) random integer in [1,u] using randomi(1,u,seed) needs one multiplication operation and at most u subtraction operations for a modular operation.

Then Procedure Random_Selection(j, T) requires one assignment to initialise u. Its first i-for-loop (for initialising B) requires T assignments. The second takes

$\sum_{k=T-j}^{k=T} k$

subtractions and j multiplications to generate R. The third requires j assignments to *selection*, and at most

$\sum_{k=T-j}^{k=T} k$

assignments to B for the j executions of the k-for-loop. Adding, we find that the time complexity of Random_Selection(j, T) is

$$\begin{aligned} C_T(Random-Selection(j,T)) &= 1+T+(\sum_{k=T-j}^{k=T} k)+j+(\sum_{k=T-j}^{k=T} k) \\ &= j(2T+j+2)+T-1. \qquad \text{(F.3)} \end{aligned}$$

In the worst case, there are $T{=}N$ packets in tempbuff, $j{=}N$ of which could be transferred into central buffer. Then the total number of scalar operations required for Procedure Random-Selection(N, N) is $N(3N + 2) + N$ -1.

Adding the number of operations required to execute procedure Reception we find

$$\begin{aligned} C_T(Reception\ procedure\ of\ sCA/B) &= (N+3)+(N(3N+2)+N-1) \\ &\quad +C_T(Plan\_Receptions(H)\ of\ sCA/B) \\ &= 8N^2+15N+2. \qquad \text{(F.4)} \end{aligned}$$

Adding the time complexity of the Transmit procedure with the Receive procedure we obtain the time complexity of sCA/B:

$$\begin{aligned} C_T(sCA/B) &= C_T(Transmission\ procedure\ of\ sCA/B) \\ &\quad + C_T(Reception\ procedure\ of\ sCA/B) \\ &= 8N^2 + 18N + 2. \end{aligned} \tag{F.5}$$

During each time slot, each ordinary station executes a station Transmission procedure (2 comparisons), a Reception procedure (1 comparison), and an Arbitration procedure (N+1 comparisons, 2(N-1) assignments, and N-1 additions) invoked within Reception. The maximum computational complexity of the sCA/B MAC protocol for ordinary stations is therefore $4N + 3$ scalar operations. The network computational complexity of a sCA/B network with N stations is therefore

$$\begin{aligned} C_N(sCA/B) &= (8N^2 + 18N + 2) + N(4N + 3) \\ &= 12N^2 + 21N + 2 \end{aligned} \tag{F.6}$$

F.2 Time Computation Complexity of optCA-MRS*

As mentioned, the CA-STAR networks differs from the other networks in that ordinary stations are released from tasks associated with conflict resolution so their MAC protocols have lower computational complexity than those of stations in other networks. The time computational complexity of CA-STAR protocols should therefore be represented by the time complexity of the MAC protocol of the CA station.

According to the optCA-MRS* protocol, during one time-slot, the Transmission Procedure of optCA makes N comparisons (with f_i, i=1,2, ...,N) to determine which packet (if any) should be transmitted from buffer module Q_i. Thus

$$C_T(Transmission\ Procedure\ of\ optCA\ according\ to\ optCA-MRS^*) = N. \tag{F.7}$$

The Reception Procedure of optCA (according to optCA-MRS*) makes N comparisons (with p_i, i=1,2, ...,N) to determine if the packet from S_i needs

to be received. Then it executes the $P = P^+$ and $F = F^+$ assignments. We assumed that these matrix assignments are implemented by rotating the role of P with P^+, and F with F^+, during every time-slot, and that the read accesses to them (for the above mentioned comparison operations) alternate from time-slot to time-slot. Since the role of the memory (or registers) allocated for storing the matrices alternates deterministically, no pointer nor assignment operations are required for these matrix "assignments". Lastly, the Reception Procedure makes one call to Procedure MRS*(W) to update M , P^+, and F^+. Thus

$$C_T(ReceptionProc.of optCAaccordingtooptCA-MRS^*) = N+C_T(ProcedureMRS^*(W)) \quad \text{(F.8)}$$

Step 1 of MRS*(W) requires N comparisons and N assignments for initialising P^+.

Step 2 of MRS*(W) requires $2N^2$ comparisons and N^2 assignments (to generate D).

Step 3 of MRS*(W) requires $2N^2$ additions[1] and $2N$ assignments (to generate G and H).

Step 4 requires at most $2N$ comparisons and N assignments to find the smallest element in G[2], N comparisons and N assignments for finding all elements in G equal to the smallest value, and 1 multiplication and N subtractions to choose one of them at random. Up to $2N$ comparisons and N assignments are needed to find the index of the smallest element in H, and one assignment is needed for recording the index in q.

Step 5 requires at most N comparisons, and N assignments. If the random selection of one element from the qth row of D is executed, then $(p \geq q)$ and the random selection of one of the smallest elements of H would not be executed. Hence the operations needed for one of the two possible random selection tasks should be counted.

Step 6 requires 4 comparisons, $4N$+6 assignments, $2N$ subtractions, and 1 logical operation, assuming that the most complex computation branch was executed.

[1]Increments to the control loop variable are not counted, since they are interpreted at compile time (not executed at runtime), to give the relative addresses of the operands of the addition instructions

[2]It was assumed in [CHEN91] that this task required only $2N$-1 operations. Perhaps it had been assumed that the values of the smallest value during the search is stored in a register, so "assignments" to it were not counted as an assignment operation

Step 7 requires at most N comparisons. (We assume associative testing is not available, to be on the pessimistic side).

Step 8 requires $2N$ comparisons, $2N$ assignments, N additions and N subtractions.

Steps 4 to 7 are repeated at most N times. Adding, we get

$$\begin{aligned} C_T(\textit{Reception proc. of optCA under optCA} - MRS^*) &= N + N(18N + 13) + 5N^2 + 10. \\ &= 23N^2 + 24N \qquad \text{(F.}\end{aligned}$$

Adding the number of operations required to execute the Transmission and Reception procedures of optCA-MRS* we get

$$\begin{aligned} C_T(optCA - MRS^*) &= C_T(\textit{Transmission Proc. of optCA under optCA} - MRS^*) \\ &\quad + C_T(\textit{Reception Proc. of optCA under optCA} - MRS^*) \\ &= 23N^2 + 25N. \qquad \text{(F.10)} \end{aligned}$$

During each time-slot, each ordinary station executes a station Transmission procedure (2 comparisons) and a Reception procedure (one comparison, and one assignment). Thus,

$$C_T(\textit{Station MAC protocol of optCA} - MRS^*) = 4. \qquad \text{(F.11)}$$

The network computational complexity of an optCA-MRS* network with N stations is therefore

$$\begin{aligned} C_N(optCA - MRS^*) &= 23N^2 + 25N + 4(N) \\ &= 23N^2 + 29N. \qquad \text{(F.12)} \end{aligned}$$

F.3 Time Computational Complexity of CF-WDMA Networks

In an CF-WDMA (request-schedule-then-transmit) network [CHEN91], [CHEN92], each station S_i, i=1,...,N, maintains the following variables :

- Backlog Matrix $B = [b_{j,k}]_{N \times N}$. $b_{j,k}$ indicates[3] the number of packets at S_j destined for S_k.
- Backlog Indication Matrix $D = [d_{j,k}]_{N \times N}$. $d_{j,k}$=1 if S_j has at least one packet destined for S_k; $d_{j,k}$=0 o.w..
- Transmission Matrix $M = [m_{j,k}]_{N \times N}$. $m_{j,k}$=1 if S_j should transmit a packet to S_k; $m_{j,k}$=0 o.w..

Every station is required to establish a global view of all packets waiting for transmission in all stations (i.e. the B matrix), during every time-slot. During each time-slot, each station use B and the Maximum Remaining Sum (MRS) scheduling algorithm to compute M, the conflict-free transmission schedule for the following slot. $m_{j,k}$=1 if S_j should transmit a packet to S_k during the next slot.

Propagation delay was assumed to be zero in [CHEN91] and [CHEN92], thus if $m_{k,i} = 1$ (S_k is scheduled to transmit a packet to S_i during the next time slot) then S_i should receive the packet from S_k (on λ_k) during the next time slot. To allow comparison, we assume that the propagation delay is not zero. We assume that all stations are a slots from the star coupler. Thus if $m_{k,i} = 1$ during time-slot t (i.e. S_k is scheduled to transmit a packet to S_i during $t + 1$) then S_i should receive the packet from S_k (on λ_k) during $t + 1 + 2a$.

Accordingly, each station S_i also maintains :

- Next Transmission Index, T. At the beginning of t, T=j if S_i should transmit the packet at the head of its jth FIFO queue to station S_j. T=0 o.w.
- Planned reception Index, I, $I \in \{1, ..., 2a\}$. I is initialised to 0
- Planned reception matrix, $P = [p_i]_{2a \times 1}$, $i \in 1, ..., 2a$. At the end of t, p_I= j if during t+1 S_i should receive the packet transmitted by S_j (on λ_j) ; p_I, =0 o.w.
- Destination address matrix, $H = [h_i]_{N \times 1}$ is an $N \times 1$ array of station addresses. h_i=address received from the ith mini-slot of the current control slot.

[3]Previously B denoted buffer capacity. Let the scope of this new meaning of B be restricted to this sub-section.

According to the CF-WDMA protocol, every station performs three tasks during every time-slot. Wlog, consider the tasks of S_i during t.

Task 1: Request Broadcasting. When a new packet arrives to S_i for transmission to S_j, S_i writes the packet's destination address, i.e. j, on the ith mini-slot of the outgoing control slot. Every control slot is divided into N mini-slots, where mini-slot i is for Task 1 of S_i. The new packet is added to the tail of the jth FIFO transmission queue of S_i's transmit buffer.

Task 2: Scheduling. This task determines which packets should be transmitted by stations during t+1 and hence which channel S_i should receive from during t+1+2a. It consists of 4 steps.

1. Update B. During t, S_i receives all mini-slots and stores their contents in H. S_i updates B by incrementing its elements corresponding to new packet transmission requests (indicated by H) and decrementing elements corresponding to packets transmitted from stations (indicated by M). Specifically, this step can be performed by executing the following statements :

   // add new packet transmission requests (indicated by H) to B
   for j=1 to N **do**
    if ($a_j \geq 0$) **then** $b_{j,h_j} = b_{j,h_j} + 1$;
   // subtract M from B, then initialise M to zero
   for j=1 to N **do**
    for k=1 to N **do**
     if ($m_{j,k} == 1$)
      then { $b_{j,k} = b_{j,k}$ - 1 ; $m_{j,k} = 0$; }

   This step requires at most N comparison, N assignments and N additions to execute the first j-for-loop; and N^2 comparisons, $2N^2$ assignments and N^2 subtractions for executing the nested j/k for-loops. The total number of scalar operations required is then $4N^2 + 3N$.

2. Find D from B as follows :

   for j=1 to N **do**
    for k=1 to N **do**
     if ($b_{j,k} \geq == 1$) **then** $d_{j,k} = 1$; **else** $d_{j,k} = 0$;

   At most N^2 comparisons and N^2 assignments are needed for this step.

3. Find new Transmission Matrix, M. Use the MRS algorithm to find a new M from D. M then indicates the packet(s) to be transmitted by all stations during $t + 1$.

MRS attempts to maximise the number of packets that can be transmitted by stations during the next time-slot, subject to i) the conflict-free constraint and ii) that each station can transmit at most one packet during the next time-slot. Formally, MRS tries to find M with maximum rank subject to:

(a) $\sum_{k=1}^{k=N} m_{kj} \leq 1$, $1 \leq j \leq N$, and

(b) $\sum_{j=1}^{j=N} m_{kj} \leq 1$, $1 \leq k \leq N$.

The complexity of the MRS algorithm has already been analysed in [CHEN91], and was shown to be

$$C_T(MRS) = 12N^2 - 2N. \qquad \text{(F.13)}$$

4. Scheduling packet transmission and reception. This step can be performed by executing the following statements :

$I = I + 1$;
if ($I \geq 2a$) **then** I=1 ;
// The following is a more efficient way of evaluating mod(I, $2a$),
// given that we know the behaviour of I
$T = I$ -1 ;
if (T == 0) **then** T=$2a$;
for j=1 to N **do**
{ **if** ($m_{i,j}$== 1) **then** T=j ; **else** T=0 ;
// Schedule a reception $2a$ time-slots from now
if ($m_{j,i}$== 1) **then** P_I=j ; **else** P_I=0 ; }

This step requires at most $2N$+2 comparisons, $2N$+4 assignments, and one addition and one subtraction operation.

Task 3. Transmit and/or receive packets scheduled in the previous slot

This task requires at most 2 comparisons (with T and P_I).

Adding the number of scalar operations required, we find

$$\begin{aligned} C_T(CF-WDMA) &= (4N^2+3N)+(2N^2)+(12N^2-2N)+4N+8+2 \\ &= 18N^2+5N+10 \qquad \text{(F.14)} \end{aligned}$$

F.4 Time Computational Complexity of the DT-WDMA Protocol

In a DT-WDMA [CHEN90] network, all stations execute the same DT-WDMA MAC protocol during every time-slot. Refer to section 2.4.1 on page 42. The DT-WDMA protocol is based on the "detect-and-retransmit" principle. It is made of a Transmission procedure, a Reception procedure, and an Arbitration procedure.

The Transmission procedure defines the steps to be followed for packet transmission, success detection, and for the retransmission of lost packets. The Reception procedure specifies the steps for packet reception, and for detecting destination conflicts. If a destination conflict is detected, then the Arbitration procedure is called.

First, consider the time complexity of the Transmission procedure. Packets in the transmit buffer of each station are logically organised into a special FIFO queue. Each new packet is tagged with its arrival time (one assignment) and a status flag which is initialised to *waiting* (one assignment). The packet in the transmit buffer queue which 1) has its status set to *waiting* and 2) has the earliest arrival time amongst the *waiting* packets, is chosen for transmission. Hence packets are chosen according to a "conditional-FIFO" discipline. At most one packet would be chosen and scheduled for transmission during one time-slot. When, S_i schedules a packet for transmission to S_j during t, it transmits its destination address, i.e. j, and the arrival time of the packet, in the ith mini-slot of the outgoing control slot. Then the packet is transmitted during $t+1$ on λ_i, and the status of (the copy of) that packet in the transmit buffer is set to *outstanding* (one assignment).

The outcome of the transmission is known to S_i after a period equal to $2a_i$ time-slots, where a_i is the normalised propagation delay from S_i to the star coupler. During $t+2a_i$, S_i could deduce whether the packet transmitted during $t+1$ would be received or lost. The procedure used by S_i for deducing the outcome of the transmitted packet was named *Arbitration*. If the transmission was successful, S_i finds the copy of the packet in its queue and removes the packet from its buffer. Otherwise the status flag of the packet is changed from *outstanding* to *waiting*.

The complexity of the Transmission procedure varies depending on the method for finding the *outstanding* packet that was transmitted $2a_i$ time-slots ago (whose outcome could therefore be deduced using procedure Arbitration). We assume that this queue search is supported by special buffer access instruc-

tions. We also assume that the packet in the transmit buffer queue with status set to *waiting* and with the earliest arrival time, is found using a special buffer access instruction. Neither contribute to the DT-WDMA protocol's computational complexity.

As mentioned, during every time-slot t, S_i invokes an Arbitration procedure for deducing the outcome of packets transmitted during $t - 2a_i - 1$. According to Arbitration [CHEN90], a packet is successfully received if it is the only packet arriving to the destination (i.e. no destination conflict occurred). Alternatively, if the packet is involved in a destination conflict, then the packet is received only if it had the *largest delay* of all the packets involved.

The complexity of DT-WDMA also depends on how a destination behaves when a destination conflict occurs, and two or more packets involved have the same largest delay. In this case we assume that the station randomly selects one of the packets with the largest delay for reception. We assume that Arbitration is implemented as follows. During t the station receives all mini-slots in the incoming control slot. The addresses from the mini-lots are stored in a

- Destination address matrix, $H = [h_i]_{N\times 1}$. H is an $N \times 1$ matrix of station addresses. $h_i=j$, i=1,2, ...,N, indicates that S_i had sent a packet to S_j in the following data slot on λ_i.

Similarly let the values of packet generation times (at source stations) received from the mini-slots be stored in the

- Packet delay matrix, $D = [d_i]_{N\times 1}$. D is an $N \times 1$ array. $d_i=k$ indicates that S_i had sent a packet on λ_i, and the generation time of the packet equals k.

Denote the destination of the packet transmitted by S_i during $t-2a_i-1$ by X. Thus $X=h_i$. After receiving the addresses in the mini-slots into H and D, the transmission procedure calls Arbitration to determine whether the packet was received.

Procedure DT-WDMA Arbitration (H,D,N,i)
// H,D, are as defined above. N is the number of stations, i is the address of S_i as usual
// returns 1 if S_i's packet is successful (received by destination);
// returns 0 if S_i's packet is lost due to a destination conflict.
Local Variables

```
  (register) integer j ;
  integer X ;
  W =[w_k]_{N×1}, k=1,...,N
  integer n_W ;
  M =[m_k]_{N×1}, k=1,...,N ;
  integer n_M ;
  integer Earliest ;
  L =[l_k]_{N×1}, l=1,...,N ;
  integer n_L ;
Begin
  n_W = 0 ;
  n_M = 0 ;
  Largest = 0 ;
  X = h_i ;
  for j=1 to N do
      if ( h_j== X ) then { n_W=n_W+1 ; w_{n_W}=j ; M_{n_W}=d_j } ;
  // Now, elements in w_1, w_2, ..., w_{n_H} are addresses of all stations that transmitted
  // a packet to station S_X

  // Find the earliest generation time (largest delay) of packets destined for S_X
    for j=1 to n_W do
        if ( m_j ≤ Earliest ) then Earliest = m_j ;

  // Find packet(s) transmitted by one of S_{w_k}, k=1,2, ...,N, with the earliest
  // generation time (largest delay)
    for j=1 to n_W do
        if ( m_j == Earliest ) then { n_L=n_L+1 ; l_{n_L}=w_j ; } ;

  // Select at random one of the stations which transmitted a packet to S_X and whose
  // packet has the largest delay. Return 1 if its own packet was selected for reception.
  if ( l_{randomi(1,n_L,rand())} == i ) then return(1);
    else return(0) ;
End ;
```

The execution of Arbitration involves at most : 4 assignments to initialise n_W, n_M, *Earliest*, and X; N comparisons, $3N$ assignments, and N additions to execute the first j-for-loop for generating W; N comparisons, N assignments to find *Earliest* (the second j-for-loop) ; N comparisons, N assignments, and N additions to execute the third j-for-loop for generating L; and 1 multiplication and N subtractions for randomly choosing one element from $\{l_1, l_2, ..., l_{n\_L}\}$. Hence the time complexity of procedure DT-WDMA

Arbitration is

$$C_T(Arbitration\ procedure\ of\ DT-WDMA) = 11N + 5. \qquad \text{(F.15)}$$

If 1 is returned from Arbitration, the packet was successful and S_i purges the copy of the packet in its transmit buffer. We assume that the copy of this packet stored in the buffer of S_i can be found and deleted using a special buffer access instruction. However, if 0 was returned, the packet was lost. S_i locates the copy of the packet in its transmit buffer, and changes its status from *outstanding* to *waiting*. Thus the packet is eligible to be considered for (re)transmission, following the above Transmission rules. The time complexity of the Transmission procedure of DT-WDMA therefore is

$$\begin{aligned} C_T(Transmission\ procedure\ of\ DT-WDMA) &= 3 + (11N + 5) + 1 \\ &= 11N + 9 \qquad \text{(F.16)} \end{aligned}$$

A Reception procedure is executed by S_i during every slot as well. It determines which packet (and channel) S_i should receive from during the next time-slot, based on values of the H and D matrices defined above. As mentioned H and D are generated from information in the mini-slots received during the current time-slot.

During t, the Reception procedure of S_i uses H and D to determine whether any packets are destined for itself during t+1. If a destination conflict occurred, a procedure almost identical to Arbitration is called to select one for reception. The others packets would not be received (i.e. would be lost). The index of the channel carrying the selected packet is assigned to a variable. Hence

$$C_T(Reception\ procedure\ of\ DT-WDMA) = 11N + 5 + 1. \qquad \text{(F.17)}$$

Adding gives

$$C_T(DT-WDMA) = 22N + 15. \qquad \text{(F.18)}$$

F.5 Time Computational Complexity of the DAS Protocol

The Dynamic Allocation Scheme (DAS) protocol [CHIP93] allocates transmission rights to stations on a packet-by-packet basis during every time-slot, generally following the "request-schedule-then-transmit" idea of CF-WDMA. The algorithm used by DAS for producing the conflict-free transmission schedules is called the Random Scheduling Algorithm (RS). During every time-slot, every station use RS to select the packets that should be transmitted from stations during the next time-slot. Like CF-WDMA, all stations execute the same algorithm. Stations are assumed to use a common random number generator seed. In this way, all stations will mutually arrive at the same conclusion. The RS algorithm is executed at the beginning of every time-slot. Its time computational complexity has been already analysed in [CHEN94] to be

$$C_T(DAS\ Protocol) = \sum_{j=1}^{j=N} \sum_{k=1}^{k=N} (N-j+2)(N-k+2)(N+j-1). \quad \text{(F.19)}$$

F.6 Time Computational Complexity of the Hybrid-TDM Protocol

The Hybrid-Time Division Multiplexing protocol for WDM star networks combines the fixed-transmission-rights method, with the request-schedule-then-transmit method (e.g. CF-WDMA, DAS) for resolving destination conflicts. Refer to section 2.3.3 and [CHIP93] for a description.

Permissions to transmit packets during N slots of each frame are assigned to specific source-destination pairs. These permissions are fixed (i.e. same during every frame, regardless of demand). Deviating from a pure fixed assignment scheme, after every n time-slots, where $n = N/M$, one slot is left "open". Transmission permissions during such "open" time-slots are determined dynamically using a DAS like scheme. Thus all stations execute the RS algorithm to determine permissible packet transmissions during open slots. RS is modified to account for the packets that have already been transmitted during non-open slots.

The advantage of the HTDM protocol is that the RS algorithm is invoked only once every N/M slots, $M < N$. Hence the time computational complexity can be estimated by

$$
\begin{aligned}
C_T(HTDM) &= \frac{M}{N+M} C_T(DAS) \\
&= \frac{M}{N+M} \sum_{j=1}^{j=N} \sum_{k=1}^{k=N} (N-j+2)(N-k+2)(N+j-1) \quad \text{(F.20)}
\end{aligned}
$$

Appendix G

References

[ACAM89] A.Acampora, "An overview of lightwave packet networks," IEEE Network Magazine, vol.3., no.1, pp.29-41, Jan. 1989.

[ACAM91] A.Acampora and I.Shah, "Multihop lightwave networks: A comparison of store-and-forward and hot-potato routing," Proc. IEEE INFOCOM'92, pp.10-19, 1991.

[ACAM92a] A.Acampora, S.I.A.Shah, "Multihop lightwave networks: a comparison of store-and-forward and hot-potato routing," IEEE Trans. on Commun., vol.40, no.6, pp.1082-90, 1992.

[ACAM92] A.Acampora, S.I.A.Shah, and Z.Zhang, "Performance analysis of hot-potato routing for multiclass traffic in multihop lightwave networks," Proc. IEEE INFOCOM'92, pp.644-655, 1992.

[ACAM94] A.S.Acampora,"The scalable lightwave network," IEEE Communications Magazine, vol. 32, no.12 pp. 36-42, 1994.

[ADDI90] Addie, R.G. "NetCAD System Architecture". Telecom Australia, Switched Networks Research Branch, Branch Paper 190, NetCAD Report 4, Feb. 1990.

[AHMA89] H.Ahmadi and W.Denzel, "A survey of modern high-performance switching techniques", IEEE J. Select. Areas Commun., vol. 7, no. 7, Aug. 1989, pp. 1091-1103.

[AJMO90] Ajmone-Marsan, M., G.Balbo, G.Bruno, and F.Neri. "TOPNET: a Tool for the Visual Simulation of Communication Networks". IEEE J. on selected Areas in Commun., Vol.9, No.9, Dec.1990, 735-1747.

[ASGA89] Asgarkhani, M., and K.Pawlikowski. "Simulation Studies of TDMA-

Reservation Protocol". Proc. 8th Phoenix Int. Conf. on Computers and Communications, Scottsdale, USA, March 1989,195-200.

[BAIO90] A.Baiocchi, M.Carsi, M.Listanti, G.Pacifici, A.Roveri, R.Winkler, "The ACCI access protocol for a twin bus ATM metropolitian area network", Proc. IEEE INFOCOM'90, pp 165-174, 1990.

[BALH89] Bal, H.E., J.G.Steiner, , and A.S.Tanenbaum, "Programming Languages for Distributed Computing Systems", ACM Computing Surveys, vol.21, No.3, Sept.1989.

[BANE91] S.Banerjee, B.Mukherjee, and D.Sarkar, "Heuristic algorithms for constructing near-optimal structures of linear multihop lightwave networks," Tech. report no. CSE-91-29, Division of Computer Science, University of California, Davis, 1991.

[BANE92] S.Banerjee and B.Mukherjee, "An efficient and fair probabilistic scheduling protocol for multi-channel lightwave networks", Proc. ICC'92, Ju,.1992.

[BANE92] S.Banerjee and B.Mukherjee, "Incorporating continuation-of-message information, slot reuse, and fairness in DQDB networks," Computer Networks and ISDN Systems, vol. 24, no.2 pp. 153-69, April 1992.

[BANE92] S.Banerjee, B.Mukherjee, and D.Sarkar, "Heuristic algorithms for constructing near-optimal structures of linear multihop lightwave networks", Proc. IEEE INFOCOM'92, pp.671-680, 1992.

[BANE94] S.Banerjee, B.Mukherjee, D.Sarkar, "Heuristic algorithms for constructing optimized structures of linear multihop lightwave networks," IEEE Trans. on Commun., vol.42, no.2-4, pp.1811-1826, 1994.

[BANN90a] A. Bannister, L. Fratta, and M.Gerla, "Topological design of the wavelength-division optical network," Proc. IEEE INFOCOM'90, pp1005-1013, 1990.

[BANN90b] A. Bannister, and M.Gerla, "Design of the wavelength-division optical network," Proc. IEEE ICC'90, pp962-967, 1990.

[BARR95] R.A.Barry, and P.A.Humblet, "Models of blocking probability in all-optical networks with and without wavelength changers" Proc. IEEE INFOCOM'95, pp.402-412, Boston MAssachusetts, 1995.

[BIRR84] Birrell A.D., and B.J.Nelson, "Implementing Remote Procedure Calls". ACM Trans on Comput. Systems, vol.2, Feb.1984, 39-59.

[BOGI93b] K.Bogineni, P.W.Dowd, "Analytical modeling of WDM media access protocols," Proc. of the Twenty-Sixth Hawaii International Conference

on System Sciences, vol.1, pp.266-75, 1993.

[BOGI93c] K.Bogineni, P.W.Dowd, "Impact of propagation delay on media access protocol performance for star-coupled WDM local area networks," Proc. MILCOM '93, vol.1, pp.298-302, 1993.

[BOGI93] K.Bogineni, K.M.Sivalingam, and P.W.Dowd, "Low-complexity multiple access protocols for wavelength-division multipleded photonic networks," IEEE JSAC, vol.11, no.4, pp590-603, May 1993.

[BRAC90] C.Brackett, "Dense wavelength division multiplexing networks: Principles and applications," IEEE J. Select. Areas Commun., vol. 8, no. 6, Aug. 1990, pp. 948-964.

[BREW95] G.B.Brewster, and M.K.Vernon, "The fairness of DQDB networks with slot reuse," Proc. IEEE INFOCOM '95, pp. 1154-63 vol.3, IEEE Comput. Soc. Press, 1995.

[BRIA90] Brian N.B., T.E.Anderson, and E.D.Lazowska, "Lightweight Remote Procedure Call". ACM Trans. on Comput. Sys., vol.8, Feb.1990, 37-55.

[CASA90] S.Sacale, V.Catania, A.LaCorte, "ATM and Adaptsation layer in a DQDB MAN", 7th ITC Seminar, New Jersey, October 1990.

[CAVA91] J.A.Cavailles et al., "First digital optical switch based on INP/GaInAsP double heterostructure waveguides", Electronic Letters, vol.27, no.6, 1991.

[CHAN93] S.-H.G.Chan, H.K obayashi, "Performance analysis of Shufflenet with deflection routing," Proc. GLOBECOM '93, vol.2, pp. 854-9 , 1993.

[CHAW92] M.J.Chawki, R.Auffret, E.Le Coquil, P.Pottier, L.Berthou, ; H.Paciullo, J.Le Bihan, "Two-electrode DFB laser filter used as a wide tunable narrowband FM receiver: tuning analysis, characteristics and experimental FSK-WDM system," Journal of Lightwave Technology, vol.10, no.10, pp.1388-1397, 1992.

[CHEN90] M. S. Chen, N. R. Dono, and R. Ramaswami, "A media-access protocol for packet- switched wavelength division multiaccess metropolitan area networks," IEEE J. Select. Areas Commun., vol. 8, no. 6, Aug. 1990, pp. 1048-1057.

[CHEN91] Chen M. and T-S. Yum. "A conflict-free protocol for optical WDMA networks". Proc. IEEE GLOBECOM'91, IEEE Press, 1991, pp. 1276-1281.

[CHEN92] M.Chen and T.Yum. "Buffer sharing in conflict-free WDMA networks". Proc. IEEE INFOCOM'92, IEEE Press, 1992, pp. 664-670.

[CHEN94] M.Chen, N.Georganas and O.Yang. "A fast algorithm for multi-channel/port traffic assignment". Proc. IEEE GLOBECOM'94, IEEE Press, 1994, pp. 96-100.

[CHEN95] D.Chen, C.L.Fincher, D.A.Hinkley, R.A. Chodzko, T.S.Rose, R.A. Fields, "Semimonolithic Nd:YAG ring resonator for generating CW single-frequency output at 1.06 mm," Optics Letters, vol. 20, no.11 pp. 1283-1285, 1995.

[CHEU89] K.Cheung, D.Smith, J.Baran, abd B.Heffner, "multiple channel operation of an integrated acousto-optic tunable filter," Electronic letters, vol.25, pp.375-376, 1989.

[CHI95] Y. Cui; M. Zhang, "Acoustooptic tunable filter transmission characteristic analysis for incident cone of light," Proc. of the SPIE - The International Society for Optical Engineering.

[CHIN95] V.R.Chinni, T.C.Huang, P.K.A.Wai, C.R.Menyuk, Simonis, C.J., "Performance of field-induced directional coupler switches," IEEE Journal of Quantum Electronics, vol.31, no.11, pp.2068-2074, 1995.

[CHIP92] R.Chipalkatti, Z.Zhang, and A.Acampora,"High-speed communication protocols for optical star networks using WDM", Proc. INFOCOM'92, 1992.

[CHIP93] Chipalkatti R., Z.Zhang and A.S.Acampora. "Protocols for optical star-coupler network using WDM: Performance and complexity study". IEEE J. Select. Areas Comm., vol.11, no. 4, May 1993, pp. 579-589.

[CHLA87] I.Chlamtac and A.Ganz, "Toward alternative high speed networks: The SWIFT architecture," Proc. IEEE INFOCOM'87, pp.1102-1108, 1987.

[CHLA88a] I.Chlamtac, " Toward alternative high speed networks: The SWIFT architecture," Proc. INFOCOM'87, pp.1102-1108, 1987.

[CHLA88a] I.Chlamtac, " Toward alternative high speed networks: The swift architecture," Proc. INFOCOM'87, pp.1102-1108, 1987.

[CHLA88b] I.Chlamtac, "Frequency-time controlled multichannel networks for high speed communication," IEEE Trans. on Communications, pp.430-440, Apr.1988.

[CHLA88] I.Chlamtac and A.Ganz, "Channel allocation protocols in Frequency-time controlled high speed networks"," IEEE trans. on Commun., vol.36, pp.430-440, Apr.1988.

[CHLA90] I.Chlamtac and W.Franta, "Rationale, directions, and issues surrounding high speed networks," Proceedings of the IEEE, vol.78, no.1, pp.94-

120 Jan.1990.

[CHLA91] I. Chlamatac and A. Fumagalli, "QUADRO-Stars: High performance optical WDM star networks," Proc. IEEE GLOBECOM'91, pp. 1224-1229, 1991.

[CHLA94] I.Chlamtac and A.Fumagalli, "Quadro: A solution to packet switching in optical transmission networks," Computer networks and ISDN systems, vol.26, pp.945-963, 1994.

[CONT91] M.Conti, E.Gregori, and L.Lenzini, "Methodological approach to an extensive analysis of DQDB performance and fairness", IEEE J. Select. Areas Commun., vol. 9, no. 1, Jan. 1991, pp. 1048-1057.

[CONTI90] M.Conti, E.Gregori, and L.Lenzini, "DQDB under heavy load:performance evaluation and fairness analysis," Proc. IEEE INFOCOM'90, pp.313-320, 1990.

[CROS93] I.R.Croston, ; A.D.Carr, ; N.J.Parsons, ; S.N.Radcliffe, ; L.J.St.Ville, "Lithium niobate electro-optic tunable filter with high sidelobe suppression," Electronics Letters, vol. 29, no.2, pp. 157-159, 1993.

[DAVI94] P.Davids, T.Meuser, and O.Spaniol, "FDDI: status and perspectives," Computer Networks and ISDN Systems, vol. 26, pp. 657-677, 1994.

[DEPR93] De Prycker M. Asychronous Transfer Mode,

[DIAS88] A.R Dias, R.F. Kalman, J.W. Goodman, and A.A.Sawchuk, "Fiber optic crossbar switch with broadcast capability," Opt. Eng., vol.27, no.11, pp.955-960, Nov. 1988.

[DRAG89] C.Gragone, "Efficient N X N ster couplers using Fourier optics," Jour. Lightwave Technology, vol.7, pp.479-489, 1989.

[DRAV91] Dravida, S., M.A.Rodrigues and V.R.Saksena. "Performance comparison of high-speed networks". Proc. ITC'13 (Copenhagen, Denmark), June 1991, pp.967-973.

[DYKE88] D.Dykeman and W.Bux, "Analysis and tuning of the FDDI media access control protocol", IEEE J. Select. Areas Commun., vol. 6, no. 6, Jul. 1988, pp. 997-1010.

[FIJI88] Fijiwara et al, "A coherent photonic wavelength-division switching system for broadband networks," Proc. of the 14th Euro. conference on optical communications, ECOC'88, pp.139-142, 1988.

[FOOT95] E.M.Foo and T.G.Robertazzi, "A distributed global queue transmission strategy for a WDM optical fiber network," Proc. IEEE INFO-

COM'95, pp.154-161, 1995.

[FORG95] F.Forghieri, A.Bononi, and P.R.Prucnal, "Analysis and comparison of hot-potato and single-buffer deflection routing in very high bit rate optical mesh networks," IEEE Trans. on Communications, vol.43, no.1, pp. 88-98, 1995.

[FRAT81] L.Fratta, F.Borgonovo, and F.A.Tobagi, "The Express-net: a local area communication network integrating voice and data," Performance of data communication systems, pp.77-88, G.Pujolle Editor, Amsterdam, The netherlands, Pub. North-Holland, 1981.

[FRAT94] L.Fratta, F.Borgonovo, J.Bannister, AND M.Gerla, "Routing and admission control in the multihop wavelength-division optical network," Computer networks and ISDN systems, vol.26, pp985-1005, 1994.

[FUJI90] Fujimoto, R.M. "Parallel Discrete Event Simulation". Comm. ACM, no.10, Oct.1990, pp.31-53.

[GANZ89] A. Ganz and I. Chlamatac, "Path allocation access control in fiber optic communication systems," IEEE Trans. on Computers., vol.c-38,no.10, pp.1372-1382, Oct. 1989.

[GANZ91] A. Ganz and Z. Koren, "WDM Passive star - protocols and performance analysis," Proc. IEEE INFOCOM 91, pp. 991-1000, 1991.

[GANZ92] A.Ganz, Y. Gao, "A time-wavelength assignment algorithm for a WDM star network," Proc. IEEE INFOCOM '92, vol.3, pp.2144-2150, 1992.

[GANZ93] A.Ganz and B.Li, "A packet-switched WDM passive optical star based metropolitan area network", Proc IEEE INFOCOM'93, pp 57-63, 1993.

[GANZ93] A.GAnz, W.Gong, and X.Wang, "Wavelength assignment in multihop lightwave networks," Proc. IEEE INFOCOM'93, pp.1367-1374, 1993.

[GANZ94a] A.Ganz, X.Wang, "Efficient algorithm for virtual topology design in multihop lightwave networks," IEEE/ACM Transactions on Networking, vol.2 no.3, pp.217-25, 1994.

[GANZ94c] A.Ganz, W.Gong, and X.Wang, "Wavelength assignment in multihop lightwave networks," IEEE Transactions on Communications, vol.42, no.7, pp.2460-2469, 1994.

[GANZ94] A.Ganz, ; Y.Gao, "Time-wavelength assignment algorithms for high performance WDM star based systems," IEEE Transactions on Communications, vol.42, no.2-4, pp.1827-1836, 1994.

[GEHA88] Gehani,N.H., and W.D.Roome. "Rendezvous Facilities: Concur-

rent C and the Ada Language". IEEE Trans. on Software Eng., vol.14, 11, Nov. 1988, 1546- 1553.

[GLAN91a] B.Glance and M.J.Karol, "Protection-Against-Collision (PAC) optical packet network: Architecture and performance ," Proc. IOOC-ECOC'91, pp.745-748, Sep.1991.

[GLAN91] B. Glance, U. Koren, C. A. Burrus, and J. D. Evankow, "Discretely-tuned n-frequency laser for packet switching applications based on WDM," Electronic letters, vol. 27, no. 15, pp. 1381-1383, Jul. 1991.

[GLYN91] Glynn, P.W. and P.Heidelberger. "Analysis of Parallel Replicated Simulations under a Completion Time Constraint". ACM Trans. on Modelling and Computer Simulation, no.1, 1991, pp.2-23.

[GOOD89] M.S.Goodman, "Multiwavelength networks and new approaches to packet switching," IEEE Communications Magazine, pp.27-35, Oct.1989.

[GOOD90] M.S.Goodman, J.L.Gimlett, H.Kobrinski, M.P.Vecchi, and R.M.Bulley, "The LAMBDANET multiwavelength netowrk: Architecture, applications, and demonstrations," IEEE JSAC, vol.8, pp.955-1004, Aug.1990.

[GREE91] P.E.Green, "The future of fiber-optic computer networks," IEEE Computer, vol.24, no. 9, pp.78-87, Sep.1991.

[GREE93] P.E.Green, Fiber Optic Networks, Prentice Hall, 1993.

[GUOY94] D.Guo, Y.Yemini, Z.Zhang, "Scalable high-speed protocols for WDM optical star networks, " Proc. IEEE INFOCOM '94, vol.3, pp.1544-51, 1994.

[GUST88] Gustafson, J.L. "Reevaluating Amdahl's Law". Comm. ACM, vol.31, no.5, May 1988, pp.532-533.

[HABB87] I. M. I. Habbab, M. Kavehrad, and C. W. Sundberg, "Protocols for very high-speed optical fiber local area networks using a passive star topology," J. Lightwave Technol., vol. LT-5, no. 12, pp.1782-1794, Dec. 1987.

[HANS72] Hansen P., "Structured multiprogramming," Commun. of the ACM, vol.15, no.7, pp.574-578, 1972.

[HEID81] P.Heidelberger and P.D.Welch, "A spectral method for confidence interval generation and run length control in simulations", Comm. of the ACM, 1981, pp.233-245.

[HEID86] Heidelberger, P. "Statistical Analysis of Parallel Simulation". Proc. 1986 Winter Simulation Conf., IEEE Press, 1996, 290-295.

[HEID88] Heidelberger, P. "Discrete Event Simu-lation and Parallel Simula-

tion: Statistical Properties". SIAM J. Scientific and Statistical Computing, vol.9, 1988, 1114-1132.

[HENR89] P.S.Henry, "High-Capacity lightwave local area networks," IEEE Comm. Mag., pp.20-26, Oct. 1989.

[HINK93] I.Hinkov, V.Hinkov, E.Wagner, "Low power integrated acousto-optical tunable filters in first telecommunication window," Electronics Letters, vol. 30, no. 22, pp. 1884-1885, 1994.

[HLUC91] M.G.Hluchyj and M.J.Karol, "Shuffle Net: An application of generalized perfect shuffles to multihop lightwave networks," Proc. IEEE INFOCOM'88, pp.379-390, Mar. 1988.

[HLUC91] M.Hluchyj and M.Karol, "ShuffleNet: An application of generalized perfect shuffles to multihop lightwave networks", Jour. Lightwave Technlogy, vol.9, no.10, Oct.1991, pp.1386-1396.

[HOAR78] Hoare, C.A.R. "Communicating sequential Processes". Comm. ACM, vol. 24, Feb. 1978, pp.75-83.

[HOARE74] Hoare C.A.R. "Monitors: an operating system structuring concept," Comm. of the ACM, vol.17, no.10, pp.149-557, 1974.

[HOOG77] Hoogendoorn C.H. "A general model for memory interference in multiprocessors". IEEE Trans. on Computers, Oct. 1977, pp.998-1005

[HUM93] P. Humblet, R.Ramaswami, and K.Sivarajan, "An efficient communication protocol for high-speed packet-switched multichannel networks," IEEE JSAC, vol.11, pp.568-577, May.1993.

[HUNT95] D.K.Hunter, I.Andonovic, "Architectures for synchronous optical TDM switching employing semiconductor laser amplifiers," IEE Proceedings-Optoelectronics, vol.142, no.3, pp.132-142, 1995.

[HWAN95] W.Y. Hwang, J. Kim, T. Zyung, M.Oh, S.Y. Shin, "Postphotobleaching method for the control of coupling constant in an electro-optic polymer directional coupler switch," Applied Physics Letters, vol. 67, no. 6, pp.763-765, 1995.

[IEEE90] IEEE: Proposed Standard 802.6. Distributed Queue Dual Bus (DQDB), subnetwork of a metropolitan area network (MAN)", Oct. 1990.

[INES95] J.Iness, S.Banerjee, B.Mukherjee, "GEMNET: a generalized, shuffle-exchange-based, regular, scalable, modular, multihop, WDM lightwave network," IEEE/ACM Transactions on Networking, vol.3, no.4, pp.470-476, 1995.

[INUK79] Inukai T. "An efficient SS/TDMA time slot assignment algorithm".

IEEE Trans. on Comm., vol. 27, no.10, Oct.1979, pp.1449-1455.

[IRSH92] M. I. Irshid and M. Kavehrad, "A WDM cross-connected star topology for multihop lightwave networks," J. Lightwave Technol., vol. 10, no. 6, pp. 828-835, Jun. 1992.

[JAGE94] D.Jager, A.Stohr, O.Humbach, "High-speed nin-waveguide modulator and switch," LEOS '94. Conference Proceedings, vol.1, pp. 236-237.

[JEON95] H.B.Jeon and C-K.Un, "Contention-based reservation protocols in multiwavelength optical networks with a passive star topology," IEEE Trans. on Commun., vol.43, no.11, pp.2794-2802, 1995.

[JIAB93] F.Jia, and B.Mukherjee, "The receiver collision avoidance (RCA) protocol for a single-hop WDM lightwave network," Journal of Lightwave Technology, vol.11, no. 5-6, pp.1053-1065, 1993.

[JIAM92] F. Jia and B. Mukherjee, "The receiver collision avoidance (RCA) protocol for a single- hop WDM lightwave network," Proc. IEEE ICC'92, June 1992.

[JOHN87] M. Johnson, "Proof that timing requirements of the FDDItoken ring protocol are satisfied", IEEE Trans. on Commun., vol. COM-35, no. 6, Jun. 1987, pp. 620-625.

[KARO91a] M.J.Karol and B.Glance, "Performance of the PAC optical packet netwwork," Proc. IEEE Globecom'91, p, M.J.Karol and B.Glance, "A collision-avoidance WDM optical star network," Comp. Networks and ISDN systems, vol.26, pp.931-943, 1994.

[KARO91b] M.J.Karol, and S.Z.Shaikh, "A simple adaptive routing scheme for congestion control in ShuffleNet multihop lightwave networks," IEEE Journal on Selected Areas in Communications, vol.9 , no.7 pp. 1040-1051, Sept. 1991.

[KARO91c] M.J.Karol and B.Glance, "Performance of the PAC optical packet netwwork," Proc. IEEE Globecom'91, pp.1258-1263, Dec.1991.

[KARO94] M.J.Karol and B.Glance, "A collision-avoidance WDM optical star network," Comp. Networks and ISDN systems, vol.26, pp.931-943, 1994.

[KAUD87] Kaudel, F.J. "A Literature Survey on Distributed Discrete Event Simulation". Simuletter, vol.18, no.2, June 1987, pp.11-21.

[KAZO90] L.Kazovsky, "Optical signal processing for lightwave communications networks," IEEE J. Select. Areas Commun., vol. 8, no. 6, Aug. 1990, pp.973-981.

[KAZO93] L.G.Kazovsky, C.F.Barry, M.J.Hickey, C.A.Noronha, and P.T.Poggiolini, "A multi-Gbit/s optical LAN utilizing a passive WDM star: towards an experimental prototype," Proc. IEEE INFOCOM'93, pp.48-56, 1993.

[KIM90] B.Kim, "Packet delays in the IEEE 802.6 DQDB protocol", Proc. IEEE Int. Conf. on Commun, ICC'90, pp.1692-1696.

[KIMP92] C.Kim, C-M.Park, J.Youn, "Slot reuse based on address comparison in the DQDB protocol", Proc. ISADS'93, International Symposium on Autonomous Decentralized Systems, pp.262-265, IEE Comput. Soc. Press.

[KIVI91] Kiviat, P.J. "Simulation, Technology and the Decision Process". ACM Trans. on Modeling and Computer Simulation, vol.1, No.2, April 1991, pp.89-98.

[KLEI79] Kleijnen, J.P.C. "The Role of Statistical Methodology in Simulation". In: "Methodology in Systems Modelling and Simulation", B.P.Zeigler et al. (eds), North-Holland, Amsterdam, 1979.

[KOBR87] H.Kobrinski, "Demonstration of high capacity in the LAMBDANET architecture: a multiwavelength optical network," Elect. Letters, vol.23, pp.824, 1987.

[KOBR89] H. Kobrinski, and K-W Cheung, "Wavelength-tunable optical filters: applications and technologies," IEEE Commun. Magazine, pp.53-63, Oct . 1989.

[KOVA94a] M.Kovacevic, M.Gerla, J.Bannister, "On the performance of shared-channel multihop lightwave networks," Proceedings IEEE INFOCOM '94, vol.2, pp. 544-51, IEEE Comput. Soc. Press, 1994.

[KOVA94] M.Kovacevic and M.Gerla, "HONET: An integrated services wavelength division optical network," IEEE INFOCOM'94, pp.1669-1674, 1994.

[KOVA95a] M.Kovacevic and A.Acampora, "On wavelength translation in all optical networks" Proc. IEEE INFOCOM'95, pp.413-422, Boston MAssachusetts, 1995.

[KOVA95] M.Kovacevic and M.Gerla, "A new optical signal routing scheme for linear lightwave networks," IEEE Trans. on Communications, vol.43, no.12, pp.3004-3014, 1995.

[KREU92a] Kreutzer, W. "The Role of Graphic and Animation in Simulation Software". New Zealand Operational Research, Proc. 28th Annual Conf. (Christchurch, August 1992), Operational Research So. of New Zealand, Wellington, 1992, pp.183-190.

[KREU92] Kreutzer, W., and Y.Sundralingam. "QN-Movie: A Toolbox for

Hierarchical Composition & Animation of Queueeing Networks". Proc. Summer Computer Simulation Conf. (Reno, July 1992), SCi Inc., 1992.

[KUNG92] H.T.Kung "Gigabit local area networks: A systems perspective", IEEE Communications Magazine, Apr. 1992, pp79-89.

[KUWA94] S.Kuwano, O.Ishida, N.Shibata, H.Ishii, T.Kitoh, "Switching performance of an optical FDM/TDM crossconnect at the data rates of 622 Mb/s and 2.5 Gb/s," Proc. ECOC '94. 20th European Conference on Optical Communication, vol.2, pp.575-578.

[LABO90] J.P.Labourdette and A.Acampora, "Wavelength agility in multihop lightwave networks," Proc. IEEE INFOCOM'90, pp.1022-1029, 1990.

[LABO91] J Labourdette and A.Acompora, "Logically rearrangable multihop lightwave networks," IEEE Trans. on Commun. vol.39, pp.1223-1230, Aug.1991.

[LEE89] T. P. Lee, and C-E. Zah, "Wavelength-tunable and single-frequency semiconductor lasers for photonic communications networks," IEEE Commun. Magazine, pp.42-52, Oct. 1989.

[LEE91a] T-P. Lee, "Recent advances in long-wavelength semiconductor lasers for optical fiber communication," Proc. of the IEEE, vol.7, no.3, Mar.1991, pp.253-272.

[LEE91] H.W.Lee, "Protocols for multichannel optical fiber LAN using passive star topology," Electronic Letters, vol.27, no.17, pp.1506-1507.

[LEE95] Wei Yu Lee; Jin Shin Lin; Kun-Yi Lee; Wei-Ching Chuang, "SSFLC optical directional coupler switch with a short device length: a proposal," Jour. of Lightwave Technology, vol. 13, no.11, pp.2236-2243, 1995.

[LEEL95a] K-C.Lee, and V.O.K.Li, "Optimization of a WDM optical packet switch with wavelength converters" Proc. IEEE INFOCOM'95, pp.423-429, Boston MAssachusetts, 1995.

[LICH92] C-S.Li, M.S.Chen, and F.Tong, "Architecture and protocol of a passive optical packet switched metropolitan/wide area network using WDMA", Proc. Globecom'92, pp.1891-1895, 1992.

[LIN89] Y-K. Lin, D. Spears, and M.Yin, "Fiber-based local access network architectures," IEEE Communications magazine, pp.64-72, Oct.1989.

[LITO92] C-S Li, F.Tong, K.Liu, and D.Messerschmitt, "Channel capacity optimization of chirp-limited dense WDM/WDMA systems using OOK/FSK modulation and optical filters", Jour. of Lightwave Technology, vol.10, no.8, Aug.1992, pp.1148-1161.

[LIU95] J.Y. Liu and K.M. Johnson, "Analog smectic C* ferroelectric liquid crystal Fabry-Perot optical tunable filter," IEEE Photonics Technology Letters, vol: 7, no. 11, pp.1309-1311, 1995.

[LOPE93] M.Lopez-amo, J.M.Lopez-higuera, M.A.Muriel, "An electro-optically tunable filter for wavelength demultiplexing," International Journal of Optoelectronics, vol. 8, no.1, pp. 1-5, 1993.

[MAO95] C.C.Mao, D.McKnight, K.M.Johnson, "High-speed liquid crystal on silicon spatial light modulators," Proc. International Conference on Optical Computing, pp. 539-542, 1995.

[MART93] P.Martini, "Update on 802.6 ongoing work on DQDB," in Interworking in Broadband Networks, S.Rao (Ed), IOS Press, pp.77-87, 1993.

[MEHR90] N.Mehravari, "Performance and protocol improvements for very high speed optical fiber local area networks using a passive star topology," Jour. of lightwave tech., vol.8, no.4, pp.520-530, Apr.1990.

[MISR86] Misra, M. "Distributed Discrete- Event Simulation". ACM Computing Surveys, no.1 March 1986, pp.39-65.

[MUKH91] B. Mukherjee, "Architectures and protocols for WDM-based local lightwave networks," Department report CSE-91-32, Aug. 1991, Dept. of Electrical Engineering and Computer Science, University of California, Davis.

[MUKH92] B. Mukherjee, "WDM-Based local lightwave networks part I: Single-hop systems," IEEE Network, vol.6, no.3, pp.12-27, May 1992.

[MUKH92] B.Mukherjee and C.A.Bisdikian, "A journey through the DQDB network literature," Performance Evaluation, vol: 16 no.1-3 pp.129-58, Nov. 1992

[MUKH92] B.Mukherjee and F.Jia, "Bimodal throughput characteristics of a single-hop WDM systm," Proc. OFC'92, Feb. 1992.

[MUKH92] B.Mukherjee, "WDM-based local lightwave networks. Part I: Single-hop systems," IEEE Network, pp.12-27, May 1992.

[MUKH92] B.Mukherjee, "WDM-based local lightwave networks. Part II: Multihop systems," IEEE Network, pp.20-32, May 1992.

[MUKH93] F.Jia, and B.Mukherjee, "The receiver collision avoidance (RCA) protocol for a single-hop WDM lightwave network," Journal of Lightwave Technology, vol.11, no.5-6, pp.1053-65, 1993.

[NAM92] S.Nam and C.Un, "Performance analysis of a multichannel local lightwave networkwith grouping property", Proc. IEEE INFOCOM'92, pp.656-

663.

[NELS94a] W.H.Nelson, A.N.M. Choudhury, M.Abdalla, R.Bryant, E.Meland, Niland, W. "Digital optical switches for wavelength division multiplexing with extinction ratio exceeding 20 dB at both 1.3 mm and 1.5 mm," LEOS '94. Conference Proceedings, pp.259-260, vol.2, 1994.

[NELS94] W.H.Nelson, A.N.M. Choudhury, M.Abdalla, R.Bryant, E.Meland, Niland, W. "Wavelength-independent InP/InGaAsP digital optical switches with extinction ratio exceeding 20 dB at both 1.3 mm and 1.5 mm," Proc. ECOC '94. 20th European Conference on Optical Communication, vol.2, pp.523-526, 1994.

[NOLT95] H.P.Nolting, and M.Gravert, "SYNGRAT, an electrooptically controlled tunable filter with a synthesized grating structure," Optical and Quantum Electronics, vol. 27, no. 10, pp. 887-896, 1995.

[NUMA92] T.Numai, "1.5 um phase-controlled distributed feedback wavelength tunableoptical filter," IEEE Jour. of quantum electronics, vol.28, no.6, Jun.1992, pp.1508-1512.

[OBRI95] D.C.O'Brien, D.J.McKnight, "A compact holographically routed optical crossbar using a ferroelectric liquid-crystal over silicon spatial light modulator," . Proc. of the International Conference on Optical Computing, pp.187-190, 1995.

[OGUS93] M.Ogusu, Y.Shimomura, S.Ohshima, "A thermally stable Fabry-Perot tunable filter for 1 AA-spaced high-density WDM systems," IEEE Photonics Technology Letters, vol. 5, no. 10, pp.1222-1224, 1993.

[PACH95] A.R.Pach, S.Palazzo, and D.Panno, "Delay analysis of DQDB networks with slot preuse and reuse," Computer Communications, vol. 18. no. 5, pp.338-44, 1995.

[PAPA92] G.I.Papadimitriou, D.G.Maritsas, "WDM passive star networks: receiver collisions avoidance algorithms using multifeedback learning automata," Proc. 17th Conference on Local Computer Networks, pp.688-97, 1992.

[PAWL90] Pawlikowski, K. "Steady-state Simulation of Queueing Processes: A Survey of Problems and Solutions". ACM Computing Surveys, no.2, June 1990, pp.124-170.

[PAWL91] Pawlikowski K., Yau V., "Independent replications versus spectral analysis in steady-state simulation of high speed data networks", in proc. ATRS'91, Nov. 1991, Wollongong, Australia, pp.182-190.

[PAWL92] Pawlikowski K., Yau V., An empirical comparison of sequential

estimators for output data analysis in steady state simulation of high speed data networks", in proc. Operations Research Society Conference'92, pp.175-177, Christchurch, New Zealand, 1992.

[PAWL93a] Pawlikowski K., and Yau V., "Methodology for stochastic simulation for performance evaluation of data communication networks", Final Report for Telecom Corporation of New Zealand, Wellington, New Zealand. Also Technical report no. COSC 03/93, Dept. of Computer Science, University of Canterbury.

[PAWL93b] Pawlikowski K., McNickle D., and Yau V., "Object-oriented model construction, and automated distributed simulator generation and output analysis ", Final R&D report, Australia Overseas Telecom Corporation (AOTC Telstra), Con. 7314, Also, technical report no. COSC 02/93, Dept. of Computer Science, University of Canterbury.

[PAWL94] Pawlikowski K., Yau V., and McNickle D., "Distributed stochastic discrete-event simulation in parallel time streams", Proc. 1994 Winter Simulation Conference, IEEE Press, pp.723-730.

[PENG96] Haifeng Peng; Liren Liu; Feng Wang, "Optical implementation and routing technique for a SW-banyan network," Microwave and Optical Technology Letters, vol.11, no.2, pp.90-93.

[PIER94] G.R.Pieris, G.H.Sasaki, "Scheduling transmissions in WDM broadcast-and-select networks, " IEEE/ACM Transactions on Networking, vol. 2, no. 2 pp.105-110, 1994.

[POGG94] P.T.Poggiolini and S.Benedetto, "Performance evaluation of sub-carrier encoding of packet headers in quasi-all-optical broadband WDM networks," Proc. ICC'94, pp.1681-1686, 1994.

[QINH95] C.S. Qin, G.C. Huang, K.T. Chan, and K.W. Cheung, "Low drive power, sidelobe free acoustic-optic tunable filters/switches," Elect. Letters, vol.31, no.15, pp.1237-1238, Jul. 1995.

[QUAR85] Quarterman J.S., A.Silberschatz, and J.L.Peterson, "4.2BSD and 4.3BSD as Examples of the UNIX System", ACM Computing Surveys, vol.17, Dec.1985, pp.379- 418.

[RAMA93] R.Ramaswami, "Multiwavelength lightwave networks for computer communication," IEEE Comm. Mag., pp.78-88, Feb.1993.

[REGO91] Rego, V.J. and V.S.Sunderam. "Concurrent Stochastic Simulation: Experiments with EcliPSe". Proc. Int. Conf. on Performance of Distributed Systems and Integrated Communication Networks, 1991, pp.253-271.

[REGO92] Rego, V.J. and V.S.Sunderam. "Experiments in Concurrent Stochastic Simulation: the Eclipse Paradigm". J. Parallel and Distributed Computing, vol.14, 1992, pp.66-84.

[REST94] K.Restivo, "The boring facts about FDDI", Data communications International, vol.23, no.18, pp.85-90, 1994.

[ROSSP82] W.E.Ross, D.Psaltis, and R.H.Anderson, "2D Magnetio optical spatial light modulator for signal processing," SPIE Conference, Crystal City, Arlington, VA , May 3-7, 1982.

[ROUS93] G.N.Rouskas and M.H.Ammar, "Analysis and optimization of transmission schedules for single-hop WDM networks," in Proc. IEEE INFOCOM'93, pp1342-1349, 1993.

[ROUS95] G.N.Rouskas and M.H.Ammar, "Analysis and optimization of transmission schedules for single-hop WDM networks," IEEE/ACM Transactions on Networking, vol.3 no. 2 pp.211-221, 1995.

[SCHU95] E.Schulze, W. Reden, "Diffractive liquid crystal spatial light modulators with optically integrated fine-pitch phase gratings," Proc. of the SPIE, vol. 2408, pp.113-120, 1995.

[SDUH94] G.Sudhakar, M.Kavehrad, and N.Georganas, "Access protocols for passive optical star networks," Computer Networks and ISDN Systems, vol.26, pp.913-930, 1994.

[SEMA93] G.Semaan and P.Humblet, "Timing and dispersion in WDM optical star networks," Proc. IEEE INFOCOM'93, pp.573-577, 1993.

[SENI91] J.M.Senior, J.M.McVeigh and S.D.Cusworth, "Multichannel slot reservation scheme for WDM optical fiber LAN," Electronic Letters vol.27, pp.1875-1879, 1991.

[SEVC87] K. Sevcik and M. Johnson, "Cycle time properties of the FDDI token ring protocol", IEEE Trans. Software Engineering, vol. SE-13, no. 3, Mar. 1987, pp.376-385.

[SHOU94] Shou-Kou Shao and Ming-Sheng Kao, "WDM coding for high-capacity lightwave systems," Jour. of Lightwave Technology, vol.12 no.1, pp.137-148, Jan. 1994.

[SIVA92] K.M.Sivalinggam, K.Bogineni, and P.W.Dowd, "Pre-allocation media access control protocols for multiple access WQDM photonic networks," in Proc. ACM SIGCOMM'92, pp.235-246, 1992.

[SMIT77] Smith A.J. "Multiprocessor memory organisation and memory interference". Comm. of the ACM, vol.20, no.10, Oct.1977, pp. 754-761.

[SMIT90] D.Smith, J.Baran, J.Johnson, and K-W.Cheung, "Integrated-optic acoustically-tunable filters for WDM networks," IEEE J. Select. Areas Commun., vol. 8, no. 6, Aug. 1990, pp. 1151-1159.

[STAL91] W.Stallings, Data and computer communications, Macmillan Publishing Company, 1995.

[SUDH91a] G.Sudhakar, N.Georganas, and M.Kavehrad, "Slotted Aloha and reservation Aloha protocols for very high-speed optical fiber local area networks using passive star topology," Jour, Lightwave Technology, vol.9, no.10, pp.1411-1422, Oct.1991.

[SUDH91b] G.Sudhakar, M.Kavehrad, and N.Georganas, "Multi-control channel very high-speed optical fiber local area networks and their interconnections using a passive star topology," Proc. IEEE Globecom'91, pp.624-628, 1991.

[SUDH91] G.Sudhakar, N Georgenas, and M.Kavahrad, "Slotted Aloha and reservation-Aloha protocols for very high speed optical fiber networks using a passive star topology," Jour. Lightwave Technology, vol.9, pp.1411-1422, Oct.1991.

[SUND91a] G.Sudhakar, N.Georganas, and M.Kavehrad, "Slotted Aloha and reservation Aloha protocols for very high-speed optical fiber local area networks using passive star topology," Jour, Lightwave Technology, vol.9, no.10, pp.1411-1422, Oct.1991.

[SUND91b] G.Sudhakar, M.Kavehrad, and N.Georganas, "Multi-control channel very high-speed optical fiber local area networks and their interconnections using a passive star topology," Proc. IEEE Globecom'91, pp.624-628, 1991.

[SUND91] Sunderam, V.S. and V.J. Rego. "EcliPSe: A System for High Performance Concurrent Simulation". Software-Practise and Experience, vol.21, Nov.1991, pp.1189-1219.

[TAMA95] H.Yamazaki, S.Fukushima, "Holographic switch with a ferroelectric liquid-crystal spatial light modulator for a large-scale switch," Applied Optics, vol.34, no.35, pp.8137-8143, 1995.

[TANG94a] K.W.Tang, "CayleyNet: a multihop WDM-based lightwave network," Proc. IEEE INFOCOM '94, vol.3, pp. 1260-1267, 1994.

[TANG94] K.W.Tang, "BanyanNet-a bidirectional equivalent of ShuffleNet," Jour. of Lightwave Technology, vol.12, no.11 pp.2023-2031, 1994.

[TOFF80] "The third wave", Wm Morrow, 1990.

[TSEN83] C.Tseng and B.Chen, "D-net, A new cscheme for high datarate optical local area networks", IEEE J. Select. Areas Commun., vol. SAC-1;

no. 3, Aug. 1990, pp.493-499.

[WA94] P. L-K.Wa, S.Shi, J.Pamulapati, P.Cooke, M.Dutta, A.Miller, "Integration of a GaAs/AlGaAs all-optical switch with disordered branching waveguides," Proc. LEOS '94. , vol.2, pp.261-262, 1994.

[WAGN91] Wagner, D.B. and E.D.Lazowska. "Parallel Simulation of Queueing Networks: Limitations and Potentials". Performance Evaluation Review, vol.7, no.1, May 1991, pp.146-155.

[WARR95] S.T. Warr, M.C. Parker, and R.J. Mears, "Optically transparent digitally tunable wavelength filter," Elect. Letters, vol.31, no.2, pp.129-130, Jan. 1995.

[WEN94] J-H Wen, "An efficient reuse protocol for DQDB networks," Proc. Singapore ICCS '94, vol.2, pp467-473.

[WILL90] A.Willner, I.Kaminow, M.Kuzentsov, J.Stone, and L.Stulz, "1.2 Gbps closely-spaced FDMA-FSK direct-detection star network," IEEE Photonic Tech. Letters, vol.2, no.3, pp.223-226, Mar.1990.

[WILL93] K.A.Williams, T.Q.Dam, D.H.-C.Du, "A media-access protocol for time- and wavelength-division multiplexed passive star networks," IEEE Journal on Selected Areas in Communications, vol. 11, no. 4, pp.560-567, May 1993.

[YAMA95] H.Yamazaki, T.Matsunaga, S.Fukushima, "1*1104 holographic switching with a ferroelectric liquid-crystal spatial light modulator," Optics Letters, vol. 20, no.12, pp.1430-1431, 1995.

[YAU92a] Yau V and Pawlikowski K., "Improved Nested-Threshold-Cell-Discard buffer management mechanisms", in proc. IEEE Int. Conf. on Computers, Communications, and Automation, TENCON'92, (Melbourne, Australia, Nov1992), IEEE Comm.Press, 1992, pp.820-824.

[YAU92b] Yau V. and Pawlikowski K., "A class of protocols for heavy loaded multiple-channel local area networks", in proceedings IEEE/ACM International Conference on Communications ICC'92, pp.742-748, July, 1992, in Chicago, U.S.A.

[YAU92c] Yau V. and Pawlikowski K., "ATM Overload Control: Nested Threshold Cell Discarding with Suspended Execution", in proc. of the Australian Broadband Switching and Services Symposium, ABSSS'92, pp.689-706, Melbourne, 1992.

[YAU92d] Yau V. and Pawlikowski K., "On automatic partitioning, runtime control and output analysis methodology for massively parallel simulations",

Proc. European Simulation Symposium ESS 92, pp.135-139, Dresden, Germany, Nov 1992.

[YAU93] Yau V. and Pawlikowski K., "AKAROA: a package for automating generation and process control of parallel stochastic simulation", in Proc. of the Sixteenth Australian Computer Science Conference, ed.G.Gopal et al., Australian Computer Science Communications, 1993, pp71-83.

[YAU96a] Yau V., "Automating Parallel Simulation of Telecommunication Networks", Technical Report TR-COSC 05/96, Department of Computer Science, University of Canterbury, September, 1996.

[YAU96b] Yau V., and Pawlikowski K., "A Conflict-free Traffic Assignment Algorithm using Forward Planning," IEEE INFOCOM'96, vol.3. pp.1277-1284.

[ZEIG87] Zeigler, B.P. "Hierarchical, Modular Discrete-Event Modelling in an Object-Oriented Environment". Simulation, 1987, pp.219-230.

[ZHAN91] Z. Zhang and A.Acampora, "Performance analysis of multihop lightwave networks with hot potato routing and distance-age-priorities," Proc. IEEE INFOCOM'91, pp.1012-1021, 1991.

[ZHAN94] Z. Zhang and A.Acampora, "Performance analysis of multihop lightwave networks with hot potato routing and distance-age-priorities," IEEE Transactions on Communications, vol: 42, no.8, pp.2571-2581, 1994.